



TEKNILLINEN KORKEAKOULU  
Puunjalostustekniikan osasto

Tuomas Venho

## **PRINTACCESS-MENETELMÄT DVB-MHP-YMPÄRISTÖSSÄ**

Diplomityö, joka on jätetty opinnäytteenä tarkastettavaksi diplomi-insinöörin  
tutkintoa varten Espoossa 13.8.2004.

Valvoja

Professori Pirkko Oittinen



**TEKNILLINEN KORKEAKOULU**  
**Puunjalostustekniikan osasto**

**DIPLOMITYÖN TIIVISTELMÄ**

Tekijä

**Tuomas Venho**

Diplomityön nimi

**PrintAccess-menetelmät DVB-MHP-ympäristössä**

Tiivistelmä

Osa eri medioiden lähentymiskehitystä on erilaisten digitaalisten mediapäätelaitteiden välisten kommunikaatiokäytäntöjen vakiintuminen. Painotuotteiden säilyessä tärkeänä osana kuluttajien mediakäyttöä, on tarvetta kehittää niiden ja erilaisten digitaalisten medioiden välistä kommunikaatiota ja synergiaa. Painetun median ja infocom-alueen yhdistäviä ratkaisuja kutsutaan hybridimediaksi. Tämän alan ratkaisuissa yhdistetään painotuotteen ja digitaalisen päätelaitteen ominaisuuksia siten, että syntyy uusia konsepteja näiden kahden median yhteiskäyttöön.

Vuorovaikutteisen digitaalisen television mahdollistavasta DVB-MHP-tekniikasta kaavaillaan tulevaisuuden tärkeää digitaalista mediaa. Digitaalinen televisio on muutaman vuoden kuluttua todennäköisesti vakiinnuttanut asemansa joka kodin keskeisenä medialaitteena. Vuorovaikutteisuus on keskeinen osa digitaalista televisiota ja sen eri muodot kehittyvät jatkuvasti.

Tässä opinnäytetyössä tutkitaan painotuotteesta koneellisesti luetun ja digitalisoidun informaation hyödyntämistä DVB-MHP-ympäristössä. Tutkimus keskittyy löytämään käyttökelpoisimmat menetelmät, joilla tämä informaatio saadaan siirrettyä ko. ympäristöön ja joilla voidaan ohjata vuorovaikutteista sovellusta. Tutkimus suoritettiin hankkimalla kirjallisuustutkimuksella kattavat tiedot alustan teknisistä ratkaisuista ja kehittämällä alustalle ohjelmisto, joka tarjoaa kaikki tärkeimmät tämän alueen sovellusten tarvitsemat toiminnot. Olio-ohjelmoinnin periaatteiden mukaisesti ohjelmistoa voidaan hyödyntää alueen sovelluskehityksessä.

Ohjelmistoa testattiin kehittämällä sen avulla testisovellus, jossa painotuotteesta matkapuhelimella luetut URL-osoitteet mahdollistavat katsojaa kiinnostavan lisäinformaation hakemisen osaksi televisiolähetystä. Testisovelluksessa informaation siirtämiseen matkapuhelimelta MHP-alustalle käytettiin Bluetooth-yhteyttä ja MHP-vastaanottimen paluukanavarajapintaa.

Työn valvoja

**Professori Pirkko Oittinen**

Työn ohjaaja

Professuuri

**Graafinen tekniikka**

Koodi

**AS-75**

Sivumäärä

**76**

Kieli

**Suomi**

Avainsanat

**DVB, MHP, hybridimedia, PrintAccess, bluetooth, sovelluskehys**

Päiväys

**13.8.2004**





**HELSINKI UNIVERSITY OF TECHNOLOGY**  
**Department of Forest Products Technology**

**ABSTRACT OF MASTER'S THESIS**

|  |                               |
|--|-------------------------------|
| Author<br><b>Tuomas Venho</b>  |                               |
| Title of Thesis<br><b>PrintAccess methods in the DVB-MHP environment</b>   |                               |
| <p>Abstract</p> <p>One of the central factors in the media convergence is the establishment of the means of communication between different media terminals. Since printed products have an essential role in the overall media consumption, there is a need for improving the communication and synergy between them and different digital media devices. The solutions that combine the properties of printed and infocom products are called hybrid media. The applications in this area aim to produce new ways of using these two media together.</p> <p>The DVB-MHP-technology makes possible to add interactivity to the digital television. The interactive television is forecasted to become a central media device in every household in few years. The ways of interactivity develop all the time.</p> <p>The research presented in this thesis focuses to the problem of how the information that is read automatically from a printed product and digitized can be utilized in the DVB-MHP-environment. The objective of this research was to find the most usable ways of transferring the information to that environment and using the information to give commands to the MHP application. The research consisted of studying the technical characteristics of the MHP platform and developing an application framework to provide all the essential services to applications of this area. Accordingly to the principles of object-oriented programming, this framework can be further utilized in the application development in this domain.</p> <p>The framework was tested by developing a test application that makes it possible for the user to get additional information to a television program by using a code that is read with a camera phone from a printed product. A Bluetooth connection and the return channel of the MHP platform were used in transferring the information from the mobile phone to the MHP platform.</p> |                               |
| Supervisor<br><b>Professor Pirkko Oittinen</b>   | Instructor                    |
| Chair<br><b>Graphic Arts Technology</b>  | Chair code<br><b>AS-75</b>    |
| Pages<br><b>76</b>   | Language<br><b>Finnish</b>    |
| Keywords<br><b>DVB, MHP, hybrid media, PrintAccess, Bluetooth, application framework</b>   | Date<br><b>13 August 2004</b> |

## ALKUSANAT

Tein diplomityöni Teknillisen Korkeakoulun Viestintätekniikan laboratoriossa osana TEKES:in FENIX-teknologiaohjelmaan kuuluvaa laboratorion ja VTT Median yhteistä PrintAccess-projektia. Projektin myötä minulle avautui tilaisuus tutkia digitaalisen television DVB-standardia sekä erilaisia graafisen ja digitaalisen viestinnän välisiä yhteistyömuotoja. Erityisesti DVB-standardin julkinen ohjelmointirajapinta Multimedia Home Platform ja sen mahdollistamat hybridim mediasovellukset ovat olleet keskeisenä osana tätä tutkimusta.

Ohjelmointirajapinnan ja painetusta mediasta saatavan informaation yhdistämiseen liittyvän diplomityön tekeminen sopi minulle erinomaisesti, sillä koko viestintätekniikan opiskeluni ajan minua ovat erityisesti kiinnostaneet mediaviestinnän yhdentymiskehitys - mediakonvergenssi - sekä tätä kehitystä edesauttavat teknologiat. Myös opiskeluaikana hankkimani työkokemus monikanavajulkaisemisesta ja ohjelmoinnista on tukenut hyvin diplomityöni aihealuetta.

Valmistun ajankohtana, jolloin Suomen talous toipuu vuosituhannen vaihteen nousukautta seuranneesta laskusta. Erityisesti tällaisessa taloudellisessa tilanteessa tarvitaan uusia innovaatioita ja yhteistyömuotoja tekniikan eri sektoreiden välillä. Tämän päivän diplomi-insinööreiltä vaaditaan paitsi syvällistä tekniikan tuntemusta, myös kykyä hallita kokonaisuuksia ja nähdä teknologian vaikutukset ympäröivässä yhteiskunnassa. Uskon, että opintoni ja työkokemukseni ovat antaneet minulle hyvät valmiudet tehdä työtä tällaisessa ympäristössä.

Erityiskiitokset professori Pirkko Oittiselle ohjauksesta ja kannustavasta suhtautumisesta tämän työn tekemiseen. Kiitokset yhteistyöstä ja tuesta myös kaikille PrintAccess-projektin työntekijöille ja Viestintätekniikan laboratorion väelle.

Lämpimät kiitokset kaikille läheisilleni.

Espoossa 20.5.2004

Tuomas Venho



# SISÄLLYSLUETTELO

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>JOHDANTO .....</b>  | <b>1</b>  |
| 1.1      | PAINOVIESTINNÄN JA DIGI-TV:N KOHTAAMINEN .....                 | 1         |
| 1.2      | TUTKIMUKSEN TAVOITTEET .....                                   | 1         |
| 1.3      | TYÖN RAKENNE .....   | 2         |
| <b>2</b> | <b>PRINTACCESS – VIESTINTÄVÄLINEIDEN YHTEISTOIMINTAA .....</b> | <b>3</b>  |
| 2.1      | PRINTACCESS ON HYBRIDIMEDIAA .....                             | 3         |
| 2.2      | PAINOTUOTTEET HYBRIDIMEDIAN OSANA .....                        | 4         |
| 2.3      | DIGITAALINEN TELEVISIO HYBRIDIMEDIAN OSANA .....               | 5         |
| 2.4      | PRINTACCESS-SOVELLUSTEN KEHITTÄMINEN .....                     | 5         |
| 2.4.1    | <i>PrintAccess-konseptit</i> .....                             | 5         |
| 2.4.2    | <i>Konsepteista sovelluksiksi</i> .....                        | 6         |
| 2.4.3    | <i>PrintAccess-sovellusten prototyypit</i> .....               | 7         |
| <b>3</b> | <b>DIGITAALINEN TELEVISIO .....</b>                            | <b>9</b>  |
| 3.1      | NYKYTILA .....   | 9         |
| 3.2      | DVB-STANDARDI .....  | 9         |
| 3.3      | DVB -YLEISLÄHETYS .....  | 10        |
| 3.4      | PALVELUINFORMAATIO JA LÄHETYKSEN SALAAMINEN .....              | 11        |
| 3.5      | RESOLUUTIO JA KUVASUHDE .....                                  | 12        |
| 3.6      | OLIOKARUSELLI .....  | 12        |
| 3.7      | DVB-MHP-VASTAANOTTO .....                                      | 13        |
| 3.8      | MOBIILI DVB-VASTAANOTTO .....                                  | 13        |
| <b>4</b> | <b>MHP - MULTIMEDIA HOME PLATFORM .....</b>                    | <b>15</b> |
| 4.1      | YLEISTÄ .....  | 15        |
| 4.2      | MHP-STANDARDIN VERSIOT .....                                   | 15        |
| 4.3      | MHP-ARKKITEHTUURI .....  | 17        |
| 4.4      | MHP-SOVELLUSTEN HALLINTA .....                                 | 19        |
| 4.5      | SOVELLUSTEN OHJELMOINTIKIELET .....                            | 19        |
| 4.5.1    | <i>DVB-Java</i> .....  | 19        |
| 4.5.2    | <i>DVB-HTML</i> .....  | 20        |
| 4.6      | XLET-RAJAPINTA .....   | 21        |
| 4.7      | MHP-SOVELLUSTEN GRAAFISET KÄYTTÖLIITYMÄT .....                 | 22        |
| 4.8      | SYÖTTEIDEN KÄSITTELY .....                                     | 24        |
| 4.9      | MHP-LAITTEIDEN INTERNET-YHTEYDET .....                         | 27        |
| 4.10     | MHP JA XML .....   | 27        |
| 4.11     | SOVELLUSTEN TALLENTAMINEN PÄÄTELAITTEELLE .....                | 28        |
| 4.12     | MHP:N TULEVAISUUDENNÄKYMÄT .....                               | 29        |
| <b>5</b> | <b>PRINTACCESS MHP-YMPÄRISTÖSSÄ .....</b>                      | <b>31</b> |



|       |  |    |
|-------|--|----|
| 5.1   | YLEISTÄ .....  | 31 |
| 5.2   | PRINTACCESS JA MHP-VASTAANOTIN .....                           | 31 |
| 5.3   | PRINTACCESS-KOODIT .....                                       | 33 |
| 5.3.1 | Viivakoodit .....  | 33 |
| 5.3.2 | OCR –teknologiat.....  | 34 |
| 5.4   | PRINTACCESS-KOODIEN LUKULAITTEET.....                          | 34 |
| 5.5   | MHP-PÄÄTTEEN JA LUKULAITTEEN VÄLINEN TIETOLIIKENNE .....       | 35 |
| 5.6   | TIETOLIIKENTEEN FYYSINEN TASO .....                            | 36 |
| 5.6.1 | Infrapuna .....  | 36 |
| 5.6.2 | Bluetooth.....   | 37 |
| 5.6.3 | WLAN.....  | 38 |
| 5.7   | TIETOLIIKENTEEN SOVELLUSTASO .....                             | 39 |
| 5.7.1 | Vastakkeet (sockets).....                                      | 39 |
| 5.7.2 | Yhdyskäytävät .....  | 41 |
| 5.7.3 | RMI .....  | 43 |
| 6     | SOVELLUSKEHYKSET .....   | 44 |
| 6.1   | YLEISTÄ .....  | 44 |
| 6.1.1 | Lasilaatikko- ja mustalaatikkokomponentit .....                | 45 |
| 6.1.2 | Suunnittelumallit .....  | 45 |
| 6.2   | SOVELLUSKEHYSTEN LAATIMINEN .....                              | 46 |
| 7     | PRINTACCESS SOVELLUSKEHYKSEN LAATIMINEN MHP-YMPÄRISTÖÖN .....  | 48 |
| 7.1   | YLEISTÄ .....  | 48 |
| 7.2   | PRINTACCESS-SOVELLUSKEHYKSEN VAATIMUSMÄÄRITTELY .....          | 48 |
| 7.3   | PRINTACCESS-SOVELLUSKEHYKSEN ARKKITEHTUURI.....                | 50 |
| 7.3.1 | Yleinen arkkitehtuuri.....                                     | 50 |
| 7.3.2 | PrintAccess-sovellusten käynnistäminen sovelluskehysellä ..... | 51 |
| 7.3.3 | Sovelluskehysten tilan hallinta .....                          | 52 |
| 7.3.4 | Konfiguraation hallinta ja poikkeusten käsittely.....          | 53 |
| 7.3.5 | XML-informaatio ja sen tulkitseminen.....                      | 53 |
| 7.3.6 | Käyttöliittymäkomponenttien hallinta .....                     | 54 |
| 7.4   | TIETOLIIKENNE.....   | 54 |
| 7.4.1 | PrintAccess-koodien lukeminen .....                            | 54 |
| 7.4.2 | Ohjainlaitteiden yhteyskäytännöt.....                          | 55 |
| 7.4.3 | TCP/IP –yhteydet MHP-päätelaitteelta .....                     | 55 |
| 7.4.4 | TCP/IP –yhteydet MHP-päätelaitteelle .....                     | 55 |
| 7.5   | SOVELLUSKEHYKSEN TOTEUTUS JA DOKUMENTOINTI .....               | 56 |
| 7.6   | SOVELLUSKEHYKSEN TESTAUS.....                                  | 57 |
| 7.7   | TESTISOVELLUS SOVELLUSKEHYKSELLE .....                         | 59 |
| 7.8   | SOVELLUSKEHYKSEN HYÖDYNTÄMINEN TESTISOVELLUKSESSA .....        | 63 |
| 8     | JOHTOPÄÄTÖKSET JA YHTEENVETO .....                             | 64 |
|       | LÄHDELUETTELO .....  | 66 |
|       | LIITTEET .....   | 76 |

|   |    |
|---|----|
| LIITE 1: PRINTACCESS-SOVELLUSKEHYKSEN LÄHDEKOODI .....            | 76 |
| LIITE 2: PRINTACCESS-SOVELLUSKEHYKSEN JAVADOC –DOKUMENTAATIO..... | 76 |

## SANASTO

|                         |  |
|-------------------------|--|
| <b>Abstrakti luokka</b> | Eri luokille yhteisiä ominaisuuksia sisältävä luokka, josta ei voi luoda ilmentymää / instanssia. Luokka voi käyttää abstraktiin luokkaan toteutettuja ominaisuuksia käyttämällä perintää.   |
| <b>API</b>              | <b>Application Programming Interface</b> ; Käyttöjärjestelmän tai sovelluksen tarjoama rajapinta, jonka kautta niihin voidaan tehdä ohjelmakutsuja ulkopuolelta.   |
| <b>AWT</b>              | <b>Abstract Windowing Toolkit</b> ; Kokoelma Java-kielisiä rajapintoja (kts. API), jotka tarjoavat valmiita komponentteja käyttöliittymän ohjelmointiin. Uusimpien Javan versioiden mukaisissa sovelluksissa käytetään AWT:sta kehittyneempää versiota, Swingiä. |
| <b>Bluetooth</b>        | Lyhyen kantaman radiotaajuudella toimiva yhteystapa, jolla matkapuhelimet, tietokoneet ja PDA-laitteet voivat kommunikoida keskenään langattomasti.  |
| <b>DOM</b>              | <b>Document Object Model</b> ; W3C-konsortion määrittämä spesifikaatio, jonka avulla HTML- ja XML-sivuja voidaan käsitellä sovelluksen osina.  |
| <b>DSM-CC</b>           | <b>Digital Storage Media-Command and Control</b> ; Tekniikka joka mahdollistaa sovellustiedostojen lähettämisen osana DVB-lähetystä.   |
| <b>DTD</b>              | <b>Document Type Definition</b> ; Rakenteellisen dokumentin merkintäkielen määrittely, jonka avulla dokumentti voidaan koneellisesti tulkita oikein. Webissä käytetty HTML-merkintäkieli on esimerkki DTD:stä.   |
| <b>DVB</b>              | <b>Digital Video Broadcast</b> ; Digitaalisten televisiolähetysten tekninen standardi. Standardista on omat versiot digitaalisille satelliitti-, kaapeli- ja terestialilähetyksille.   |
| <b>GUI</b>              | <b>Graphical User Interface</b> ; Tietokoneen graafinen käyttöliittymä.  |
| <b>GPRS</b>             | <b>General Packet Radio Services</b> ; Teknologia, joka mahdollistaa matkapuhelinten pakettimuotoisen langattoman dataliikenteen.  |
| <b>HTTP</b>             | <b>Hypertext Transfer Protocol</b> ; Sääntömäärittely tiedostojen lähettämiseen webissä.   |
| <b>Hybridimedia</b>     | Tutkimusala, jossa käsitellään kahden tai useamman viestintävälineen sisältöjen ja toiminnallisuuksien yhdistelmiä. Tässä työssä termillä tarkoitetaan kuitupohjaisten tuotteiden (paperi ja kartonki) ja sähköisen median erilaisia konvergensseja.             |



|                                |   |
|--------------------------------|---|
| <b>IP</b>                      | <b>Internet Protocol</b> ; Protokolla, joka määrittää säännöt informaation lähettämiseksi tietokoneelta toiselle Internetissä. Jokaisella protokollaa käyttävällä laitteella on oltava IP-osoite, jonka perusteella kone löytyy verkossa.   |
| <b>Java</b>                    | Ohjelmointikieli, joka on suunniteltu erityisesti Internetin tietoliikenneympäristöjä ajatellen. Java on olio-ohjelmointikieli, mikä tarkoittaa että sillä laadittu sovellus koostuu luokista ja niiden välisistä suhteista.  |
| <b>J2ME</b>                    | <b>Java 2 Platform, Micro Edition</b> ; Teknologia, joka mahdollistaa Java-kielisten sovellusten ohjelmoinnin pieniin päätelaitteisiin kuten matkapuhelimiin ja PDA-laitteisiin.  |
| <b>JMF</b>                     | <b>Java Multimedia Framework</b> ; Java-kielen sovelluskehys, joka mahdollistaa audion, videon ja muiden aikariippuvaisten medioiden lisäämisen Java –sovelluksiin.   |
| <b>Konkreettinen luokka</b>    | Rajapinnan toteuttava luokka, joka sisältää rajapinnassa määriteltyjen metodien toiminnallisuuden.  |
| <b>Lukulaite / lukijalaite</b> | Tässä työssä lukulaitteella tarkoitetaan kannettavaa laitetta, joka kykenee lukemaan painotuotteeseen painettua koodia ja siirtämään sen sisältämän informaation toiselle laitteelle.   |
| <b>MHP</b>                     | <b>Multimedia Home Platform</b> ; DVB-projektin laatima standardi digitaalisen television vastaanottimien väliohjelmistolle (middleware).   |
| <b>Multipleksi / Muksi</b>     | Digitaalisen television lähetyksessä käytettävä bittivirta, joka sisältää tietyn palveluntuottajan palvelut. Vastaa analogisen television kanavaa sillä erotuksella, että yhdessä multipleksissa voidaan lähettää useita kanavia. Tästä syystä multipleksiä kutsutaan myös kanavanipuksi. |
| <b>Ohjainlaite</b>             | Tässä työssä ohjainlaitteella tarkoitetaan laitetta, jolla pystytään antamaan komentoja päätelaitteelle (kts. päätelaite).  |
| <b>PPP</b>                     | <b>Point to Point Protocol</b> ; Protokolla kahden tietokoneen väliseen kommunikaatioon sarjaportin välityksellä. Tyypillisesti käytetty Internet-yhteyden muodostamiseen puhelinverkon ja modeemin välityksellä.   |
| <b>PrintAccess</b>             | Yleisnimitys menetelmille, joilla painotuotteesta koneellisesti luetulla informaatiolla ohjataan digitaalisen median päätelaitteita.  |
| <b>PSTN</b>                    | <b>Public Switched Telephone Network</b> ; Puheliikenteeseen käytettävä julkinen puhelinverkko. Nykyisin suurilta osilta kokonaan digitaalinen paitsi tapauksissa, joissa viimeistä linkkiä kuluttajalle ei ole digitalisoitu esimerkiksi ISDN- tai ADSL-tekniikalla                      |

|                      |  |
|----------------------|--|
| <b>Päätelaite</b>    | Tässä työssä päätelaitteella tarkoitetaan kuluttajan käytössä olevaa laitetta, jolla on PrintAccess-palvelun graafinen käyttöliittymä.   |
| <b>RMI</b>           | <b>Remote Method Invocation</b> ; Java-teknologia, jonka avulla oliot voivat kommunikoida keskenään tietoliikenneverkon ylitse   |
| <b>SAX</b>           | <b>Simple API for XML</b> ; Ohjelmointirajapinta, jonka avulla XML-dokumenttia voidaan käsitellä sovelluksessa. SAX on yksinkertaisempi vaihtoehto DOM:lle.  |
| <b>Singleton</b>     | Olio-ohjelmoinnin käytäntö, jossa luokasta voidaan luoda vain yksi instanssi/ilmentymä kerrallaan.   |
| <b>Sovelluskehys</b> | Ohjelmistokehityksessä uudelleenkäyttöä varten suunniteltu systeemi, joka koostuu abstrakteista luokista ja säännöistä joiden mukaan näistä luodut ilmentymät toimivat.  |
| <b>TCP</b>           | <b>Transmission Control Protocol</b> ; Sääntökokoelma, jota käytetään IP-protokollan kanssa pakettien lähettämiseen Internet-verkossa. TCP huolehtii luotettavasta pakettien käsittelystä siinä missä IP huolehtii niiden kuljetuksesta.                   |
| <b>UML</b>           | <b>Unified Modelling Language</b> ; Standardi tapa olio-ohjelmoinnin toimijoiden, luokkien ja niiden välisten suhteiden mallintamiseen.  |
| <b>URL</b>           | <b>Uniform Resource Locator</b> ; Internetissä sijaitsevan tiedoston yksilöllinen osoite. URL koostuu käytettävästä protokollasta, palvelimen domain-nimestä ja mahdollisesta tiedoston nimestä palvelimella.  |
| <b>WAP</b>           | <b>Wireless Application Protocol</b> ; Spesifikaatio, joka määrittää kommunikaatioprotokollat langattomien laitteiden kuten matkapuhelinten Internet-yhteyksille.  |
| <b>WLAN</b>          | <b>Wireless Local Area Network</b> ; langaton lähiverkkoteknologia, jonka avulla laitteet voivat kommunikoida kuten langallisessa lähiverkossa. Käyttää korkeita lyhyen kantaman taajuuksia. Eri WLAN-teknologiat on määritelty IEEE 802.11 –standardissa. |
| <b>WYSIWYG</b>       | <b>What You See Is What You Get</b> ; Ohjelma, jonka avulla on mahdollista seurata käyttöliittymän ulkoasua samalla kun sitä kehitetään.   |
| <b>XML</b>           | <b>Extensible Markup Language</b> ; Dokumentissa olevan informaation kuvaamiseksi kehitetty rakenteellinen metatietoformaatti. Kielen avulla voidaan esimerkiksi jakaa joustavasti dokumentteja tietoverkossa eri tahojen välillä.                         |
| <b>XML Schema</b>    | XML-dokumentin rakenteen ja elementtien suhteiden kuvauskieli. Schemaa voidaan käyttää dokumentin oikeellisuuden tarkistamiseksi.  |



# 1 JOHDANTO

## 1.1 Painoviestinnän ja digi-tv:n kohtaaminen

Suomen kilpailukyvyyn ja teknologiakehityksen kannalta on tärkeää, että perusteknologioiden osaajat toimivat yhteistyössä toistensa ja palveluketjun muiden toimijoiden kanssa. Yksi tämänkaltainen yhteistyömalli on hybridimediaratkaisut. Hybridimediassa perinteisempi painoviestintä yhdistyy uudempien tietoteknisten kokonaisuuksien kanssa ja syntyy uusia tuote- ja toimintamalleja. Tällaisessa yhdentymiskehityksessä tietotekniikan, tietoliikenteen ja viestinnän rajat sumenevat jatkuvasti. Hybridimediaratkaisujen tekijöiltä vaaditaan eri viestintäteknologioiden monipuolista osaamista.

Digitaalisesta televisiosta MHP-ympäristöineen kaavaillaan joka kodin multimediapäätelaitetta, jolla voidaan tulevaisuudessa käyttää perinteisen televisiosisällön lisäksi myös Internetin palveluita kuten esimerkiksi tiedon hakua ja sähköpostia. Digitaalisen television myötä siirrytään yhä selvemmin televisiosisältöjen ja lisäpalvelujen maksullisuuteen sillä nykyinen televisiotoiminta ei ole taloudellisesti riittävän kannattavaa. Lisäpalveluiden ja perinteisen televisiosisällön onnistuneen yhdistämisen kautta digitaalisella televisiolla on kaikki edellytykset asemoitua erittäin tärkeänä digitaalisena viestimenä. Näistä syistä erilaisia digitaalisen televisioon liittyviä toimintamalleja ja sovelluksia on syytä tutkia ja kehittää.

TEKES:in FENIX-teknologiaohjelmaan kuuluva, TKK:n viestintäteknikan laboratorion ja VTT Tietotekniikan yhteishankkeena keväällä 2003 käynnistynyt PrintAccess-projekti tutkii erilaisia malleja, joilla painettu ja digitaalinen media saataisiin entistä paremmin toimimaan yhteen. Projekti jakautuu erilaisten koodausmenetelmien, koodien painatustekniikoiden, koodien automaattisen lukemisen ja uusien sovelluskohteiden tutkimiseen. Osana projektia on tutkia menetelmiä, joilla painotuotteesta elektronisesti luetut koodit saadaan siirrettyä digitaalisen television vuorovaikutteiset palvelut mahdollistamaan DVB-MHP-ympäristöön.

Tämän diplomityön aiheena on edellä kuvattu hybridimedian alue: painotuotteiden ja digitaalisen television yhteistoiminnallisuus. Esimerkkeinä tällaisesta toiminnallisuudesta ovat painotuotteeseen erilaisilla painotekniikoilla painetut koodit, jotka luetaan kameralla varustetulla matkapuhelimella ja siirretään langattomasti MHP-ympäristöön. Päätelaitteessa toimii Java-teknologiaan pohjautuva MHP-sovellus, joka vastaanottaa sille lähetetyn koodin, purkaa sen tarvittaessa ja suorittaa saadun informaation perusteella sovellukseen ohjelmoituja toimintoja. Toiminnot voivat esimerkiksi olla erilaisten lisäsisältöjen hakemista ja näyttämistä tai vaikkapa televisiokanavan salauksen purkamista.

## 1.2 Tutkimuksen tavoitteet

Diplomityössä selvitetään PrintAccess-konseptien tekniset vaatimukset käyttöympäristönsä suhteen ja DVB-MHP-standardin soveltuvuus näihin vaatimuksiin. Vaatimukset selvitetään kirjallisuustutkimuksella DVB-MHP-teknologiasta ja vertaamalla teknologian mah-



dollisuuksia PrintAccess-projektissa suunniteltuihin konsepteihin. MHP-teknologian syvällisen ymmärtämisen vuoksi perehdytään DVB-standardiin myös signaalin muodostamisen, lähetyksen ja vastaanoton osalta.

DVB-MHP-standardin soveltuvuutta PrintAccess-käyttöön tutkitaan kehittämällä MHP-teknologiaa käyttävän PrintAccess-sovelluskehiksen versio 1.0, jonka avulla testataan vaadittujen ominaisuuksien toimintaa MHP-ympäristössä. Sovelluskehiksen tarkoituksena on mahdollistaa digitaaliseen televisioon liittyvien PrintAccess-tuotekonseptien kehittäminen korkean realiteetin prototyypeiksi. Sovelluskehikset toimii näin työkalupakkina, jonka avulla prototyyppien kehitys on nopeampaa ja vaatii vähemmän resursseja. Lisäksi sovelluskehiksen tarkoitus on toimia tulevaisuudessa pohjana digitaalisen television ympäristössä toimivien PrintAccess-tuotteiden kehittämiseksi.

### 1.3 Työn rakenne

Diplomityön alkuosassa paneudutaan DVB ja MHP-teknologioihin, jotta työssä luotavan sovelluskehiksen taustajärjestelmät ja tekniset rajoitukset tulevat selviksi. Tämän jälkeen keskitytään PrintAccess-kontekstiin ja selvitetään tämän alueen sovellusten toiminnalliset ja tekniset vaatimukset. Sovelluskehysten laatimista ja kehiksille tyypillisiä suunnittelukäytäntöjä käydään läpi luvussa 6. Näiden käytäntöjen pohjalta luodaan luvussa 7 PrintAccess-sovelluskehikset, johon toteutetaan edellä selvitettyt PrintAccess-vaatimukset. Vaatimusten toteutumisen perusteella arvioidaan MHP-ympäristön käyttöä ja soveltuvuutta PrintAccess-tarkoituksiin. Sovelluskehikset testataan laatimalla sen avulla PrintAccess-testisovellus. Työn johtopäätökset ja pohdinta ovat luvussa 8.

Diplomityö sisältää runsaasti tietotekniikan ja tietoliikenteen termejä sekä lyhenteitä. Joitakin termejä on suomennettu tekstin luettavuuden parantamiseksi. Suomennoksessa on käytetty Suomen Tietotekniikan liiton ylläpitämää ATK-sanakirjaa [2]. Englanninkieliset termit on selvennetty joko sanan yhteydessä tai diplomityön alussa olevassa termiluettelossa. Java-ohjelmointikielen luokista ja metodeista puhuttaessa on noudatettu tapaa, jossa diplomityössä laaditut luokat esitetään kursivoituna ilman pakettimäärettä ja ulkopuolisen tahon laatimat luokat on esitetty kursivoituna koko pakettiosoitteen kanssa (esimerkiksi *org.havi.ui.HContainer*).

## 2 PRINTACCESS – VIESTINTÄVÄLINEIDEN YHTEISTOIMINTAA

### 2.1 PrintAccess on hybridimediaa

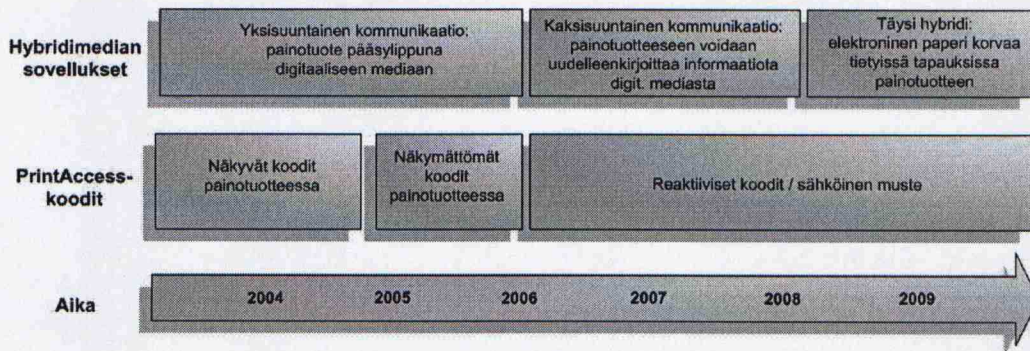
PrintAccess-termillä tarkoitetaan painotuotteiden ja sähköisen viestinnän hybridiratkaisuja, jotka mahdollistavat painotuotelähtöisesti vuorovaikutteisen viestinnän. Tällaisten ratkaisujen tarkoituksena on lisätä viestintävälineiden synergiaa ja sitä kautta painotuotteiden kilpailukykyä sekä käyttötilanteiden monimuotoisuutta, joustavuutta ja soveltuvuutta kaikille kuluttajaryhmille.

Hybridimedialla tarkoitetaan yleisesti infocom-alueen ja perinteisen paperimedian yhdistämistä /87/. Tämä on osa eri viestinten lähentymiskehitystä eli konvergenssia, joka yhdessä medioiden digitalisoitumisen kanssa on murtanut perinteisesti jakelutekniikan kautta määritellyt toimialarajat ja muuttanut viestintäalan toimintaprosesseja. Murros on ajanut yritykset toisaalta voimakkaaseen kehitykseen, toisaalta puolustamaan omia asemiaan /90/. Kehityksen suuntana on tuoda käyttäjälle yhä enemmän vaihtoehtoja mediasisältöjen kuluttamiseen mieleisillään päätelaitteilla. Tällöin mediaympäristö ja fyysinen ympäristö kohtaavat yhä paremmin /66/.

Painotuotteiden ja sähköisen median hybridiratkaisujen kehityksen voidaan hahmotella jakautuvan kolmeen vaiheeseen (Kuva 1). PrintAccess on ensimmäinen vaihe tässä kehityksessä. PrintAccess-sovellusten myötä painotuote toimii yksisuuntaisena pääsylippuna tai osoittimena digitaaliseen mediaan. Kehityksen seuraavana vaiheena on kaksisuuntainen kommunikaatio, jossa painotuotteeseen voidaan myös uudelleenkirjoittaa tietoja digitaalisesta mediasta. Lukulaitteina näkyville ja näkymättömille koodeille voidaan käyttää esimerkiksi matkapuhelimeen integroituja digitaalisia kameroita ja koodin tulkintaohjelmistojä. Kolmantena vaiheena tässä kehityksessä voidaan tietyissä tilanteissa pitää painotuotteen korvaavaa elektronista paperia tai sähkömustetta, joita sähköisten kirjojen ohella voidaan käyttää myös matkapuhelimissa, kämmentietokoneissa, televisiossa ja tietokoneen monito-reissa /23/. On kuitenkin huomattava, että elektronisen paperin kehittäminen ja käyttöönnotto saattaa kestää vielä useita vuosia /23/.

Vastaavalla tavalla voidaan PrintAccess-koodien kehitys jakaa kolmeen vaiheeseen. Ensimmäisessä vaiheessa koodit ovat näkyviä koodeja, kuten yksi- ja kaksiulotteisia viiva-koodeja. Toisessa vaiheessa näkyvä koodipainatus antaa tilaa näkymättömille koodaustavoille, kuten näkymättömälle musteelle tai esimerkiksi kuviin piilotetuille koodeille. Kolmannessa vaiheessa painotuotteessa oleva koodi on painettu sähköisellä musteella, jolloin sen sisältämää tietoa voidaan muuttaa vielä painatuksen jälkeen. /18/





**Kuva 1.** Hybridimedia-sovellusten ja access-menetelmien arvioitu kehitys lähivuosina. /18/, /23/

## 2.2 Painotuotteet hybridimediaan osana

Viestintäalan toimintaprosessien muutoksen myötä paperi- ja painoalan yritykset ovat joutuneet pohtimaan paperiviestimien kilpailukykyä suhteessa digitaalisiin medioihin /87/. Tällöin oleelliseksi kilpailukyvyn mittareiksi nousevat paperiviestintä käyttötarkoitukset ja käyttötilanteet. Kilpailukyvyn säilyttämiseksi suomalaiset paperiteollisuuden yritykset ovat jo pitkään olleet mukana useissa informaatioteknologiaa hyödyntävissä kehityshankkeissa. Näitä ovat esimerkiksi digitaalinen painatus sekä logistiikan ja pakkausteollisuuden erilaiset sovellukset.

Painotuotteiden integroiminen digitaalisen median kanssa ja näiden kahden median kaksisuuntainen kommunikaatio on hybridimediaa sivuavissa tutkimuksissa nähty keskeisenä painotuotteiden kehityksen polkuna /87/. Tällaisessa kehityskulussa painotuotteessa oleva koodi, kuten koodipainatus toimii linkkinä tai pääsylippuna digitaaliseen mediaan kuten digi-tv:n tai Internetin sisältöön. Koodipainatus voi myös mahdollistaa lehteen painetun sisällön tilaamisen käyttäjälle sopivammassa muodossa kuten äänitiedostona tai sisältöön liittyvän lisämateriaalin lataamisen digitaalisessa muodossa eri päätelaitteille /62/.

Painettu media poikkeaa sähköisestä paitsi käyttötilanteiden, myös käyttäjien osalta. Sähköinen media on usein hetkellistä ja laajoista sisällöistä valitseminen edellyttää toisten sisältöjen poissulkemista kuten esimerkiksi television tapauksessa. Monipuolisesti eri medioita käyttävät henkilöt mieltävät painotuotteet usein rajallisiksi kokonaisuuksiksi, joiden sisällöt on mahdollista ottaa haltuun. Ihmiset, jotka käyttävät runsaasti sähköisiä viestimiä, etsivät itseään kiinnostavaa tietoa useista erilaisista lähteistä, myös painotuotteista /84/. Erityisesti nuoret aikuiset kaipaavat painetulta medialta kuten sanomalehdeltä enemmän joustavuutta käyttötilanteiden suhteen. Lehteä tulisi voida lukea painetussa muodossa vaikkapa aamiaspöydässä ja kiinnostavat artikkelit tulisi olla siirrettävissä digitaaliseen mediaan esimerkiksi kuunneltaviksi työmatkalla /66/. Tällaisille kuluttajaryhmille vies-



tintävälineiden onnistuneesta yhteistoiminnallisuudesta olisi odotettavissa suurta hyötyä. Tuotekehitystyön ja käyttöliittymäsuunnittelun haasteena onkin jatkossa tutkia ja kehittää tuotekonsepteja sekä kehittää näiden pohjalta menestyksekkäitä tuotteita tälle alueelle.

## 2.3 Digitaalinen televisio hybridimedian osana

Televisiotoiminta on ollut analogisesta lähetystekniikastaan johtuen sektori, jolla digitalisoitumisen vaikutukset ovat tähän mennessä rajoittuneet lähinnä tuotantotekniikkaan. Digitaalisen television sovellusympäristöjen myötä kuluttajat voivat televisiollaan käyttää Internetin palveluja, mikä todennäköisesti muuttaa television roolin hyvin keskeiseksi digitaalisen median päätelaitteeksi. /90/

Verrattaessa digitaalisen television sovellusympäristöjä PC-laitteiden vastaaviin, on ensimmäinnittujen merkittävänä piirteenä mahdollisuus linkittyä kiinteästi television muuhun sisältöön kuten lähetettävään videoon. Digitaalisen television sovelluksissa käyttäjä voi esimerkiksi hakea tietoa lähetettävistä ohjelmista, pelata ohjelmasisältöön liittyviä pelejä tai yhdistää perinteisen televisionkatselun tiedonhakuun Internetistä /95/. Katsottaviin ohjelmiin liittyvän lisäinformaation on kuluttajatutkimuksissa havaittu kiinnostavan suurinta osaa television katselijoista /13/.

Yhteiskäyttötilanteiden lisääntyessä yksittäinen kuluttaja käyttää jatkuvasti eri viestintäkanavia ja päätelaitteita rinnakkain ja peräkkäin. Kuluttajalle mahdollistetaan yhä enemmän valinnan mahdollisuuksia itseään kiinnostavien asioiden seuraamisessa. Tällaisen kehityksen myötä myös painetun lehden lukemisen tulisi yhä paremmin tukea televisio- ja Internetsisältöjen kuluttamista /87/. Tämä tarkoittaa, että näiden viestintäalustojen tulee kyetä kommunikoidaan keskenään.

Television digitalisoituessa ja televisioympäristöön rakennettujen vuorovaikutteisten sovellusten lisääntyessä aukeaa uusia mahdollisuuksia television sisällöille ja käyttötavoille. Yksi digi-tv:n mahdollisuuksista ja ominaisuuksista on sen linkittyminen muihin päätelaitteisiin kuten mobiililaitteisiin. PrintAccessin kannalta tämä tarkoittaa, että painotuotteesta luettu koodi tulee saada siirrettyä digitaalisen television päätelaitteeseen käyttämällä jotakin päätelaitteen tietoliikennerajapinnoista. Toisaalta voi syntyä tarvetta informaation siirtämiselle digitelevision painotuotteen suuntaan. Tällainen sovellus olisi esimerkiksi kotitulostus suoraan televisiosta /98/.

## 2.4 PrintAccess-sovellusten kehittäminen

### 2.4.1 PrintAccess-konseptit

PrintAccess-konsepteilla (engl. concept) tarkoitetaan laajasti painotuotteen ja tietotekniikan yhteiskäyttöä esimerkiksi mainonnan, viihteen, uutisoinnin, televisio-ohjelmien lisäarvopalveluiden sekä julkisen sektorin palvelujen alueilla. Tutkimusten mukaan kuluttajien mielestä halutuimpia television kautta käytettäviä palveluja ovat erilaiset julkisen sektorin

palvelut. Lisäksi verkkopelaaminen, sähköposti ja elokuvien tilaaminen ja katselu kiinnostavat kuluttajia /66/. Näihin kaikkiin voidaan lisätä PrintAccess-toiminnallisuutta.

PrintAccess-projektissa suunniteltuja MHP-ympäristöä käyttäviä PrintAccess-konsepteja ovat esimerkiksi seuraavassa taulukossa (Taulukko 1) luetellut konseptit /19/.

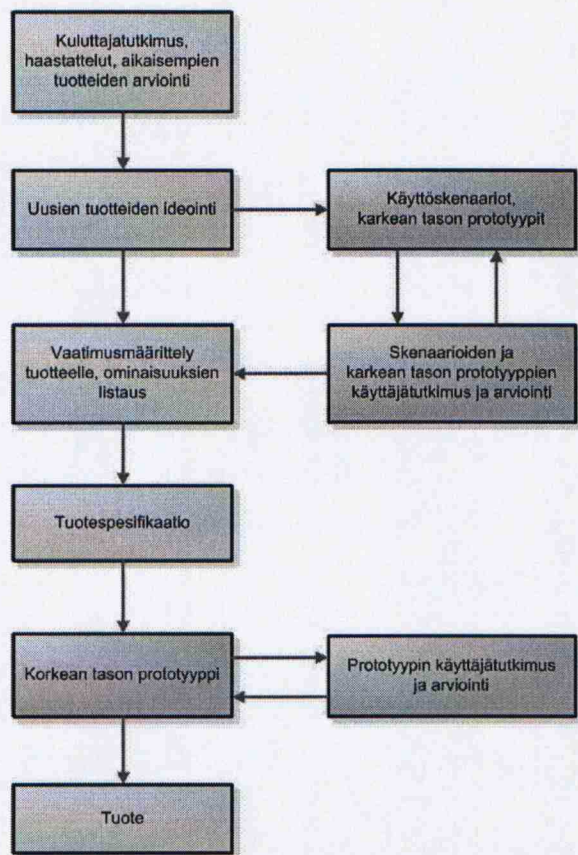
**Taulukko 1. Mahdollisia PrintAccess-konsepteja**

| PrintAccess-konsepti   | Kuvaus   |
|--|--|
| Pääsy Internetin tiettyyn URL-osoitteeseen                           | Painotuotteessa oleva koodi sisältää URL-osoitteen, jonka avulla MHP-ympäristö osaa avata suoraan tietyn www-informaation.   |
| Pay-per-view   | Painotuotteessa oleva yksilöity koodi on avain tietyn ohjelman salauksen purkamiseen. Tämä mahdollistaa ohjelmien ostamisen painotuotteen mukana tai painotuotteen sponsorimat ohjelmat  |
| Nauhoitus / etänauhitus  | Painotuotteen koodi sisältää suoraan tietyn ohjelman nauhoitukseen tarvittavat tiedot. Mikäli koodin lukulaiteella on pääsy Internetiin, voi konsepti mahdollistaa nauhoituksen käyttäjän sijainnista riippumatta.   |
| Lisäinformaatio ohjelmista / muusta katsojaa kiinnostavasta aiheesta | Koodia vastaavasti MHP-päätelaite lataa lisäinformaatiota käyttäjän luettavaksi esimerkiksi lähetettävistä ohjelmista. Lisäinformaatio voi olla myös painotuotekohtaista siten, että sitä ei ole mahdollista saada ilman oikeaa koodia. Tämä mahdollistaa esimerkiksi erilaisten TV-lehtien sponsorimat lisäinformaatiot ohjelmista. |
| MHP-kauppapaikka   | Tuotteita esittelevässä painotuotteessa on tuotteista suora koodi MHP-kauppapaikalle, josta tuote tilataan.  |
| Kyselyt / äänestykset  | Painotuotteessa oleva koodi on tietty vastaus äänestykseen. Lukemalla tämän koodin ja siirtämällä sen MHP-ympäristöön käyttäjän ääni rekisteröityy.  |

#### 2.4.2 Konsepteista sovelluksiksi

Uusien käyttöliittymätuotteiden, kuten PrintAccess-sovellusten kehitykseen voidaan soveltaa käyttäjäkohtaisen tuotekehityksen kulkukaaviota (kuva 2). Kun uusia sovelluksia (engl. application) tehdään, kootaan aluksi halutuista palveluista tuotekonsepteja tai käyttöskenaarioita. Edelleen käyttöskenaarioiden pohjalta rakennetaan sovellusten prototyyppejä ja vaatimusmäärittely toteutettaville tuotteille. Prototyypeillä tehdään käyttäjätutkimusta, jotta löydetään menestystuotteita ja saadaan tuotespesifikaatio toteutettaville tuotteille.





**Kuva 2.** Käyttäjälähtöinen tuotekehitys /110/, /67/

Yhteiskäyttöisten hybridisovellusten menestyksekkääseen toteuttamiseen tarvitaan eri välineiden käyttöympäristöjen tuntemusta sekä huolellista käyttöliittymäsuunnittelua /98/. Uudentyyppiset sovellukset vaativat myös usein kokonaan uudenlaisten käyttöliittymien tutkimista ja kehittämistä. Yksi yleisistä ongelmista uusien käyttöliittymäkonseptien kehittämisessä on, että teknologian soveltuvuutta käyttöympäristöön ei ole riittävästi selvitetty /98/, /110/. PrintAccess-projektissa tämä ongelma pyritään ratkaisemaan rakentamalla erilaisia testi- ja demoympäristöjä eri päätelaitteille ja kehittämällä tuotteiden prototyyppejä näihin ympäristöihin.

#### 2.4.3 PrintAccess-sovellusten prototyypit

Käyttäjälähtöisessä tuotesuunnittelussa konseptien visualisointi ja prototyyppien rakentaminen pyritään suorittamaan iteratiivisena prosessina, jossa edetään karkean tason (matala realiteetti) prototyypeistä korkeamman tason (korkea realiteetti) prototyyppeihin /110/. Eri

tason prototyypeillä tehdään käyttäjätestauksia, joiden tulosten perusteella laaditaan vaatimusmäärittely kehitettävälle tuotteelle.

Prototyyppiä laadittaessa yksi ratkaistava seikka on sen todenmukaisuus. Pidemmälle kehitettyjen, korkean realiteetin prototyyppien ongelmana on, että ne saattavat keskittyä liikaa jonkin tietyn teknisen yksityiskohdan toteuttamiseen. Karkean tason eli matalan realiteetin prototyypeistä puolestaan on usein vaikea saada käyttäjälle oikeaa tuotetta vastaavia /61/, /67/.

Mikäli jonkin tietoteknisen alueen sovellukset vastaavat riittävästi toisiaan on mahdollista luoda sovelluskehys, eli eräänlainen työkalupakki, jonka avulla tämän alueen sovellusten valmistusta koetetaan saada nopeammaksi /38/, /39/, /28/, /29/. Työkalupakki voidaan luoda myös testauskäyttöön, jolloin sovellusalueen prototyypit rakennetaan samalla sovelluskehyksellä kuin valmiit tuotteetkin. Tässä diplomityössä laaditaan digitaalisen television ohjelmointiympäristöön sovelluskehys PrintAccess-sovellusten prototypointiin. Tämän mahdollistamiseksi on ohjelmointiympäristöstä oltava riittävä tekninen tuntemus. Seuraavissa luvuissa perehdytään tarkemmin digitaalisen television ja sen MHP-ohjelmointiympäristön teknologiaan.



## 3 DIGITAALINEN TELEVISIO

### 3.1 Nykytila

Siirtyminen digitaaliseen televisioon on käynnissä, mutta yhä on melko epäselvää millaisia palveluja kuluttajat tulevat digi-tv-ympäristössä tulevaisuudessa käyttämään /34/. Siirryttäessä digitaaliseen televisioon siirrytään todennäköisesti myös entistä selvemmin maksulliseen televisioon. Vuorovaikutteinen digi-tv mahdollistaa niin yhteisölliset kuin interaktiiviset sovellukset sähköpostista media-arkistoihin ja ajanvietteestä koulutustarjontaan. Suuri osa näistä on lisäpalveluja, joista kuluttaja tulevaisuudessa maksaa erikseen. Lisäpalvelut ovat tärkeitä television kanavantarjoajille, sillä palveluista odotetaan saatavan tulevaisuudessa merkittäviä lisätuloja /56/. Digitaalinen televisio on kuitenkin yleistynyt melko hitaasti ja vain osa tänä päivänä myydyistä digi-tv-vastaanottimista on vuorovaikutteiset palvelut mahdollistavia DVB-MHP-vastaanottimia /7/. Tämä tarkoittaa, että digitaalisen television vuorovaikutteisten sovellusten yleistyminen kuluttajakäytössä on vielä usean vuoden päässä.

Yksi digitaaliseen televisioon vahvasti liittyvä kehityssuuntaus on Internet-yhteyksien tuominen kodin eri viihdelaitteisiin ja laitteiden välisen kommunikaation kehittyminen kodin sisäisessä lähiverkossa. Tämä edellyttää viihdelaitteisiin toteutettuja standardeja protokollia informaation siirtämiseksi. Yksi tärkeimmistä protokollista on Internetissä käytössä oleva TCP/IP, jolla pakettimuotoista dataa siirretään tietoliikenneverkon välityksellä koneelta toiselle. Myös sovellustasolle on kehitetty useita erityisesti viihdelaitteikäyttöön suunniteltuja protokollia. /91/

Kodin viihdelaitteiden kehittyvä kommunikaatioteknologia synnyttää tarvetta erilaisten kommunikaatiosovellusten käyttöön televisioympäristössä /80/,/91/. Toinen kehityssuuntaus on digitaalisten tallentimien yleistyminen televisiovastaanottimien yhteydessä ja yleinen tallennuskapasiteetin kasvaminen /34/. Tällaiset kodin viihdekeskukset, jotka toimivat televisiovastaanottimina ja joihin on toteutettu tarvittavat kommunikaatioprotokollat, mahdollistavat useita uudenlaisia hybridisovelluksia, joissa eri laitteilla kommunikoidaan televisiovastaanottimen kanssa.

### 3.2 DVB-standardi

DVB eli Digital Video Broadcasting on digitaalisen television standardi, joka on otettu käyttöön yleisesti esimerkiksi Euroopassa, useissa Aasian maissa ja Australiassa /59/. Pohjoismaissa käytössä olevaa DVB-standardia on tarkennettu NorDig-spesifikaatiossa, joka määrittää kaikille pohjoismaille yhteisen vastaanotinsuosituksen /85/. Suomessa myytävien digi-tv-vastaanottimien on siten oltava NorDig-määritysten mukaisia.

DVB-tunnuksella on kolme lisämerkintää riippuen siitä, mikä digitaalisen television lähetystekniikka on kyseessä. Eri lähetystekniikoiden lisämerkinnät ovat

- T = terrestrial (maanpäälliset lähetykset),
- C = cable (kaapelilähetykset) ja
- S = satellite (satelliittilähetykset).

Nämä lähetystekniikat eroavat toisistaan mm. niiden käyttämän modulaation osalta. Eri lähetystyyppien vastaanottimet eroavat toisistaan vastaavasti.

Edellämainittujen tekniikoiden lisäksi digitaaliselle mobiilitelevisiolle kehitetään omaa standardiaan, DVB-H:ta. Se perustuu maanpäälliseen DVB-T-tekniikkaan, mutta sitä on muokattu mahdollistamaan muun muassa vastaanottimen nopea liikkuminen ja akun säästö /105/.

3.3 DVB -yleislähetys

DVB-tekniologiassa MPEG-2-videopakkausstandardin mukainen digitaalinen informaatio siirretään yleislähetysenä (engl. broadcasting) palveluntarjoajalta vastaanottimille. DVB-yleislähetys koostuu siirtovirroista, joita kutsutaan myös kanavanipuiksi tai multiplekseiksi. Yhden kanavanipun tiedonsiirtokapasiteetti vaihtelee standardin eri lähetystekniikoiden osalta. Esimerkiksi DVB-T-tekniikan eurooppalaisessa versiossa tiedonsiirtokapasiteetti on 22,1 Mbit/s, joka tarkoittaa käytännössä 4-5 kanavaa yhdessä nipussa /96/. Muissa lähetys-tekniikoissa kanavaniput voivat olla jonkin verran suurempia /25/.

Suomen valtakunnallisessa digitelevisiossa kanavanippuja on nykyhetkellä jaettu kolme eri palveluntarjoajille (Taulukko 2). Kanavaniput on muodostettu siten, että ne ovat samansuuruisia analogisten kanavien kaistanleveyden kanssa. Kanavanippu A on varattu Yleisradion käyttöön. Kanavaniput B ja C ovat toimiluvanvaraisia. Kaikille kanavanipuille yhteistä lähetysverkkoa hallinnoi Digita Oy. /4/

Taulukko 2. Suomen digi-TV:n kanavaniput ja kanavat /4/

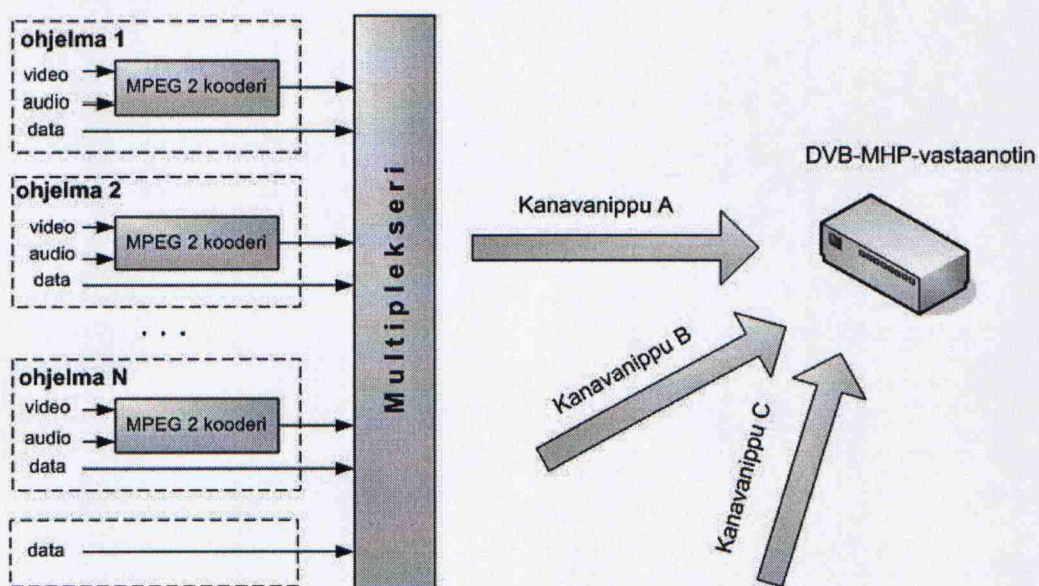
| Kanavanippu | Kanavat   |
|-------------|---|
| A           | TV1-D, TV2-D, FST, YLE24, YLE Teema                                 |
| B           | MTV3D, MTV3+, SubTV, Nelonen, Nelonen Plus                          |
| C           | CANAL+, CANAL+ Kulta, CANAL+ Sininen, Urheilukanava, VIISI, Estradi |

Yksi kanavanippu koostuu kanavista eli palveluista, jotka puolestaan koostuvat pienemistä ohjelmavirroista kuten MPEG-2 muotoisesta audiosta, videosta ja datasta. Datalla tarkoitetaan audioon ja videoon liittyvää palvelutietoa kuten ohjelmatietoja sekä vuorovai-  
kutteisten sovellusten binääri- ja tekstitiedostoja.



Jokaisella ohjelmavirralla on oma 13-bittinen PID-tunnuksensa (Program Identifier), jonka avulla se voidaan tunnistaa päätelaitteella ja sijoittaa osaksi tiettyä kanavaa. Ohjelmavirtojen lukumäärälle multipleksissa ei ole DVB-standardissa varsinaisesti rajoitusta. Ainoa rajoittava tekijä on PID:ien lukumäärä:  $2^{13}$ . Tällaisilla ehdoilla on mahdollista varata kokonainen kanavanippu pelkästään datavirralle, jolloin kanavanippu toimii koko kaistanleveydellään yksisuuntaisena datan siirtotienä. /25/

Alla olevassa kuvassa (kuva 3) on esitetty DVB-kanavanipun yleinen koostumus. Samaan nippuun kuuluvat kanavat multipleksoidaan yhteen kanavantarjoajan multiplekserissä. Multiplekseriin syötetään lähetettävien ohjelmien video MPEG-2 muodossa ja audio MPEG-1 tai MPEG-2 muodossa. Näiden lisäksi kanavanippuun multipleksoidaan sovellusdataa, palveluinformaatiota (SI), ohjelmien saanti-informaatiota (CA) ja eri formaattien synkronointi-informaatiota. Näistä video, audio ja sovellusdata kuuluvat varsinaiseen digi-tv-palveluun, eli näkyvät käyttäjälle ja kolme viimeistä ovat näkyvää palvelua tukevaa informaatiota. Synkronointi-informaatio on välttämätöntä jotta audion ja videon lähettäminen oikea-aikaisesti on mahdollista. /25/



**Kuva 3.** *Palveluiden/ohjelmien pakkaus kanavanipuksi /106/*

### 3.4 Palveluinformaatio ja lähetyksen salaaminen

DVB-kanavanippu sisältää eräänlaisen tietokannan nipussa olevista palveluista ja niihin liittyvästä palveluinformaatiosta. Nippu voi sisältää useita eri kanavia ja nämä voivat puolestaan koostua useista kamerakulmista, eritasoisista äänivirroista ja lisäarvopalveluista. DVB-palveluinformaatio (DVB-SI) on keino kertoa vastaanottimelle mitkä saatavilla olevat palvelut kuuluvat yhteen.

DVB-palveluinformaatio on taulumuotoista ja se on sisällytetty lähetyvirtaan erillisinä paketteina, joilla on oma PID-tunnuksensa. Paketit sisältävät esimerkiksi sähköisen ohjelmaoppaan eli EPG:n. Sähköinen ohjelmaopas sisältää tietoa saatavilla olevista ohjelmista, niiden sisällöstä, ajankohdista ja kestosta.

DVB-standardin mukaisesti palveluntarjoaja voi lähettää signaalia salattuna, jolloin päätelaitteella tarvitaan tekniikka salaamisen purkamiseksi. Tätä tekniikkaa kutsutaan ehdolliseksi pääsyksi palveluun (CA-Conditional Access) ja se mahdollistaa esimerkiksi tiettyjen ohjelmien katsomisen laskuttamisen. Tulevaisuudessa televisiotoiminnan ansaintalogiikka saattaaakin yhä enemmän muodostua pay-per-view-tyyppisistä toteutuksista, joissa katsoja voi tilata ohjelmia katsottavakseen esimerkiksi kännykällään /90/.

### 3.5 Resoluutio ja kuvasuhde

MPEG-2 video voi olla lähetysvirrassa enkoodattuna joko PAL- tai NTSC-resoluutioilla. Esimerkiksi Suomessa video on PAL-muodossa ja käytettävänä resoluutiona on 768x576 pikseliä /5/. Videon kuvasuhde voi olla joko 4:3 tai 16:9. Käytettävästä kuvasuhteesta ker- tova informaatio on koodattuna lähetysvirran palveluinformaatioon. Mikäli päätelaitteen kytketty televisio ei tue käytettävää kuvasuhdetta, digi-tv -pätelaitte yrittää muuttaa videon sellaiseen kuvasuhteeseen, että se voidaan näyttää /5/, /95/.

DVB-standardi mahdollistaa myös teräväpiirtolähetyksen (HDTV -High Definition TV) lähetyksen, jossa resoluutio on kaksinkertainen PAL-muotoon verrattuna. Tällainen lähety- sie kuitenkin tiedonsiirtokapasiteettia noin nelinkertaisen määrän verrattuna PAL-lähetyk- seen /93/. Tulevaisuudessa kun lähetykskapasiteettia vapautuu analogisten lähetysten pois- tuessa on mahdollista, että HDTV-lähetyksiin siirrytään mikäli niille löytyy tarpeeksi kysyntää /93/.

### 3.6 Oliokaruselli

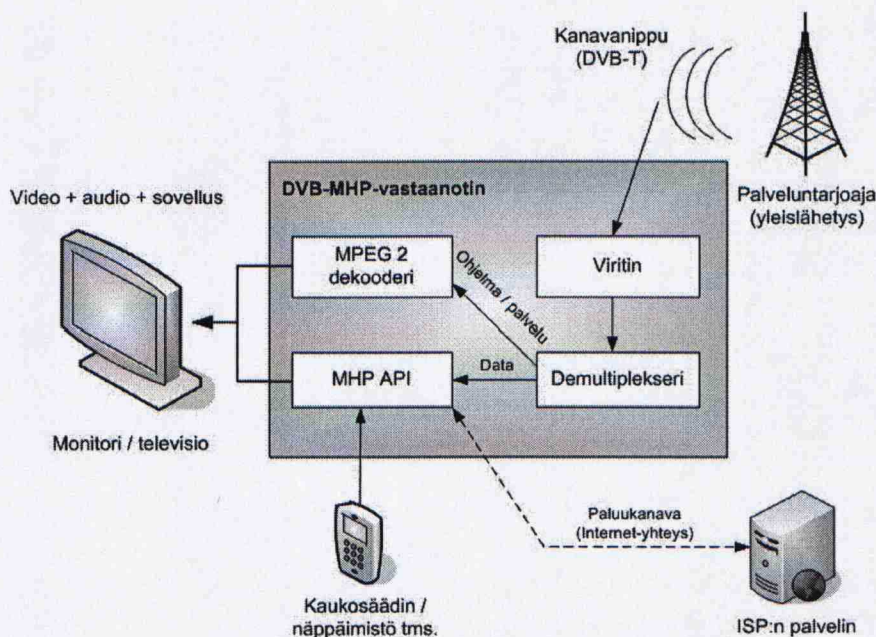
DSM-CC-tekniikalla (Digital Storage Media-Command and Control) mahdollistetaan sovellustiedostojen lähettäminen osana DVB-signaalia. Tekniikassa sovellusten tiedostot muutetaan MPEG-2 muotoon ja lähetetään vastaanottimelle. Lähettämiseen käytetään olio- karuselliä, joka on toteutettu käyttäen DSM-CC:n datakaruselliperiaatetta. Tekniikka on vastaava kuin perinteisessä tekstitelevisiossa, eli tarvittavat sovellustiedostot lähetetään lähetysvirrassa aina tietyin väliajoin.

Oliokaruselli muodostuu tiedostorakenteesta, joka jaetaan lähettämistä varten moduleihin. Yhdessä modulissa voi olla useita tiedostoja siten, että ne yhdessä muodostavat maksimis- saan 64 kilotavun kokoisen modulin. Mikäli joku lähetettävistä tiedostoista on kooltaan suurempi kuin 64 kilotavua, tulee se lähettää omana modulinaan. Tämä asettaa vaatimuksia moduleiden optimoinnille, jossa pyritään saamaan lähetettävät tiedostot mahdollisimman pieneen modulimäärään. /104/



### 3.7 DVB-MHP-vastaanotto

Kanavaniput lähetetään lähetyksessä vastaanottimelle, joka demultipleksoi kanavanipun erillisiksi kanaviksi, dekodaa MPEG-2-signaalin, käynnistää mahdolliset MHP-sovellukset ja näyttää katsojan valitseman palvelun televisioruudusta. Kaaviokuva vastaanottimen toiminnasta on esitetty seuraavassa kuvassa (kuva 4). Viritin siirtää vastaanottimen kanavanipun käyttämälle taajuudelle. Vastaanotettu signaali demoduloidaan ja siirretään demultiplekserille, joka erottelee halutun ohjelman ja mahdolliset lisäpalvelut kanavanipusta. Ohjelman video- ja audiopakettit syötetään MPEG-2 -dekooderille, joka purkaa digitaalisen informaation näytettäväksi ohjelmaksi. Lisäpalvelujen datapaketit ohjataan laitteen mikroprosessorin käsiteltäväksi ja pakettien sisältämät sovellukset suoritetaan käyttäen laitteistoon toteutettua MHP-ohjelmointirajapintaa eli API:a (Application Programming Interface).



Kuva 4. DVB-MHP-vastaanottimen toimintaperiaate [25], [106]

### 3.8 Mobiili DVB-vastaanotto

DVB-standardia kehitetään mahdollistamaan myös mobiili vastaanotto. Tätä varten standardista on laadittu versio neljännelle lähetystekniikalle, jonka nimi on DVB-H. Kyseisessä tekniikassa käytetään DVB-T-verkkoa IP-liikenteen yleislähetystekniikkaan, jolloin mahdollistetaan DVB-lähetysten mobiili vastaanotto. Tekniikka tunnetaan myös nimellä IP datacasting (IPDC). Periaatteiltaan DVB-H-tekniikka vastaa DVB-T-tekniikkaa, mutta erityispiir-

teinä DVB-H:lle on vastaanottoon kuluvan tehon pienentäminen, suurempi virhesietoisuus sekä verkon parempi kattavuus /10/, /105/.

Suomessa liikenne- ja viestintäministeriön asettama työryhmä on keväällä 2003 julkaisse-  
massaan raportissa päättänyt ehdottamaan neljännen lähetyksverkon käyttöönottoa Suomessa  
/34/. Tätä verkkoa kaavaillaan mobiilin vastaanoton mahdollistavaksi DVB-H-verkoksi.



## 4 MHP - MULTIMEDIA HOME PLATFORM

### 4.1 Yleistä

MHP eli Multimedia Home Platform on DVB-MHP-työryhmän vuodesta 1997 lähtien kehittämä avoin alusta, jolla varmistetaan että digitaalisen television vuorovaikutteiset sovellukset ovat keskenään ja eri vastaanottimien kanssa yhteensopivia. MHP:n kehittämisen päämääränä on ollut luoda horisontaalinen markkinatilanne, jossa kilpailu sisällöntuottajien, palveluntarjoajien ja laitteistovalmistajien välillä on avointa jakeluketjun kaikilla tasoilla /79/, /91/, /113/. Tällä hetkellä Euroopassa käynnissä olevassa digitaalisessa televisiotoiminnassa käytetään useita kaupallisia ja patentoituja vaihtoehtoja. Tällaisia ovat esimerkiksi MediaHighway, MHEG-5 ja OpenTV. Periaatteessa DVB-lähetysverkko tukee minkä hyvänsä vuorovaikutteiset palvelut mahdollistavan standardin käyttöä /25/, joten MHP joutuu teknologiana kilpailemaan kaikkien muiden vaihtoehtojen kanssa. Suomessa MHP on valittu vuorovaikutteisten sovellusten standardiksi valtakunnallisessa lähetysverkoissa.

MHP-kehittäjien tarkoituksena on alusta asti ollut luoda standardi ohjelmistoalusta, jonka avulla pyritään varmistamaan, että eri MHP-sovellukset ovat keskenään yhteensopivia. Jotta sovellusten lähettäminen osana televisiolähetystä mahdollistuu, on palveluntarjoajilla oltava varmuus siitä, että sovellukset toimivat kaikilla vastaanottajilla.

MHP-standardi sinänsä on riippumaton sovellusten siirtotiestä ja sen uusimmissa versioissa on tuki sekä digitaalisen television DVB-lähetystekniikoille, että Internetissä käytetylle TCP/IP-protokollalle. Pääpiirteissään MHP-standardissa määritellään /35/

- millainen on MHP-sovellus (minimivaatimukset, reunaehdot, jne.),
- miten sovellus lähetetään vastaanottiin ja
- millainen on vastaanottimissa oleva ohjelmointirajapinta (API), jonka avulla vastaanotin tulkitsee sovelluksia.

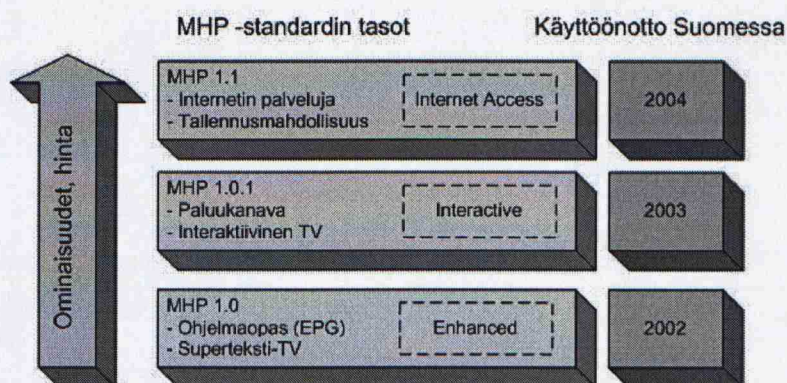
Standardista on julkaistu eri versioita ja kehitystyö on edelleen käynnissä. Standardin kehitys on ollut kuitenkin arvioitua hitaampaa /34/. Tähän ovat vaikuttaneet standardin monimutkaisuus ja standardin mukaisten laitteiden ennakoitua pienempi kysyntä /34/.

### 4.2 MHP-standardin versiot

Diplomityön kirjoitushetkellä markkinoilla olevat MHP-vastaanottimet ovat suurelta osin MHP-standardin version 1.0.2 mukaisia. Sovellusten ohjelmointiin näille vastaanottimille käytetään Java- ja HTML-kieliä, tosin versiossa 1.0.2 tuki HTML-kielelle on vielä keskenäinen. HTML-tuen keskeneräisyyden lisäksi Java-kielen merkittävämpään asemaan MHP-sovelluskehityksessä ovat vaikuttaneet sen mahdollistamat monipuolisemmat palvelut. /12/, /79/

Koska MHP on avoin alusta, periaatteessa kuka tahansa voi luoda ohjelmistoja ja sisältöä digi-tv-ympäristöön ja niitä voidaan ajaa missä tahansa MHP-yhteensopivassa vastaanottimessa. Tätä ominaisuutta pidetään MHP:n kilpailuvalttina muihin vuorovaikutteiset sovellukset mahdollistaviin teknologioihin verrattuna. Toinen avoimen järjestelmän etu on, ettei sisällöntuottajan tarvitse hankkia kallista, vain tiettyyn järjestelmään sopivaa kehitysympäristöä ja maksaa korvauksia järjestelmän tekijöille. Tällainen haittapuoli on esimerkiksi Iso-Britannian käyttöönottamassa maanpäällisessä digi-tv:ssä ja useissa Keski-Euroopan maissa käytössä olevassa OpenTV-ohjelmointiympäristössä /25/, /37/. Avoimuuden katso taankin tänä päivänä olevan yksi tärkeä edellytys ohjelmointiympäristön menestykselle /87/, /91/.

DVB-MHP:n kehitystyöstä vastaava ryhmittymä on määritellyt MHP-standardin nykyiset versiot kolmitasomallilla (Kuva 5). Kaikista kolmesta tasosta (1.0, 1.0.1, ja 1.1) on julkaistu omat spesifikaatiot. Keskimmäiseen tasoon on tehty runsaasti korjauksia ja tästä on julkaistu uudet versiot 1.0.2 ja 1.0.3.



**Kuva 5.** MHP-standardin tasot /34/, /113/

Suppein standardin versio, 1.0 eli "enhanced"-profiili, sopii edullisimpiin vastaanottimiin, joilla seurataan normaalia tv-ohjelmaa ja käytetään yksinkertaisimpia vuorovaikutteisia palveluja. "Interactive -profiili", eli version 1.0 laajennukset mahdollistavat myös paluukanavan käytön. Laajin, versio 1.1 eli internet-profiili, sisältää edellisten lisäksi myös pääsyn Internetin palveluihin ja mahdollisuuden MHP-sovellusten lataamiseen Internetistä. Tässä versiossa on myös mm. DVB-HTML-tekniikan täysi tuki sekä tuki sovellusten tallentamiselle kovalevyille, laitteiston MHP-implementaatiota laajentaville lisäohjelmille (plug-in) ja älykorttilukijoille.

Suomessa laitteistojen myynti käynnistyi siten, että MHP:n suppeinta, enhanced broadcast-profiilia tukevia laitteita on alettu myydä kaupoissa syksyllä 2002. PSTN-modeemilla toteutetun paluukanavan sisältävät laitteet tulivat myyntiin Suomessa vuoden 2003 aikana. Muilla paluukanavateknologioilla, kuten esimerkiksi ethernet-yhteyksillä



varustettuja MHP-vastaanottimia ei vielä toistaiseksi ole saatavilla. Seuraavan sukupolven, eli Internet Access-profiilin mukaisten laitteiden myynti alkaa Suomessa aikaisintaan vuoden 2004 aikana.

### 4.3 MHP-arkkitehtuuri

MHP arkkitehtuuri jakautuu pääpiirteissään kolmeen kerrokseen: laiteresursseihin, laitteiston ohjelmistoon ja sovelluksiin. Laiteresursseilla tarkoitetaan MPEG-videon prosessointia, I/O-käsittelyä ja muistin- sekä grafiikan hallintaa. Laitteiston ohjelmisto käsittää sen ohjelmakoodin, joka tarjoaa sovelluksille mahdollisuuden laiteresurssien käyttöön. Sovellukset lähetetään päätelaitteelle erilliselle datalle varatussa osassa DVB-kanavanippua. Laitteiston ohjelmisto toimii siten rajapintana laiteresurssien ja sovellusten välillä. Lisäksi laitteiston ohjelmisto sisältää erillisen sovellusten hallintaohjelman, jonka vastuulla on sovellusten käynnistäminen, resurssien allokointi ja sovellusten pysäyttäminen. /31/, /36/

MHP API on ohjelmistorajapinta, joka on laadittu digitaalisen television lähetyksen vastaanottamiseen tarkoitetun laitteiston ehdoilla vuorovaikutteisten sovellusten rakentamista varten. Tästä syystä se ei yritä kilpailla varsinaisesti esimerkiksi PC-pohjaisten ohjelmointiympäristöjen kanssa. MHP API koostuu neljästä osasta, joiden kehittämisestä vastaavat eri tahot (taulukko 3). Standardointiprosessissa päädyttiin tähän ratkaisuun, koska haluttiin käyttää mahdollisimman paljon olemassaolevia ratkaisuja /79/.

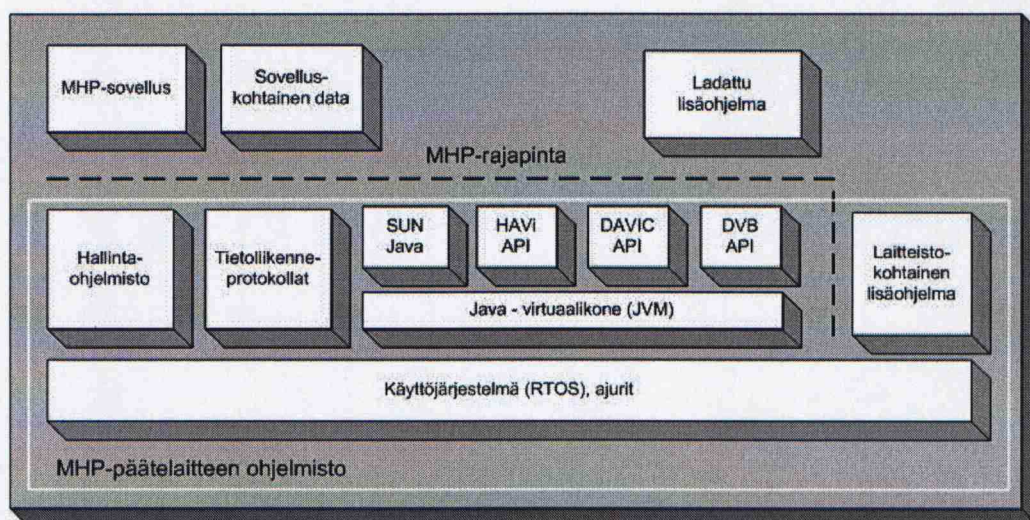
MHP-ohjelmistorajapintaa käyttäen sovellukset ja sovellusten hallintaohjelmisto voivat kommunikoida keskenään ja sovellukset saavat laitteistoresurssit käyttöönsä. Rajapinta-arkkitehtuurin avulla on pyritty varmistamaan, että digitaalisen television sovellukset toimivat erilaisissa MHP-vastaanottimissa.

**Taulukko 3.** MHP-ohjelmistorajapinnan osat /31/, /35/

| API     | Käyttötarkoitus  | Vastuullinen taho  |
|---------|--|--------------------|
| JavaTV  | Yleinen digitaalisen television toiminnallisuus. Ei ole rajoitettu DVB-standardiin vaan on käytössä myös muissa digitaalisen television standardeissa. | Sun Microsystems   |
| DAVIC   | Resurssien hallinta. MPEG-osioiden käsittely. CA-informaation hallinta   | DAVIC-organisaatio |
| HAVi    | Videon ja grafiikan käsittely, grafiikkakomponentit. Televisioympäristön erityisvaatimukset grafiikan käsittelylle                                     | HAVi-organisaatio  |
| DVB MHP | Palveluinformaation käsittely, sovellusten hallinta, oliokarusellin käsittely, paluukanavan hallinta.  | DVB-organisaatio   |



MHP:n kerrosmallinen arkkitehtuuri on esitetty seuraavassa kuvassa (Kuva 6). Alimpana kerroksellisesti on itse laitteisto, käyttöjärjestelmä ja laitteiston tarvitsemat ajurit. Digi-TV:n päätelaitteiden käyttöjärjestelmille on asetettu tiukat reaaliaikaisuusvaatimukset. Yleisesti tällaisia käyttöjärjestelmiä kutsutaan nimellä RTOS (Real Time Operating System). Nimeä käytetään useasti multimedialaitteissa korostamaan järjestelmälle asetettuja erityisen tiukkoja reaaliaikaisuuden vaatimuksia /115/.



**Kuva 6.** MHP-päätelaitteen ohjelmistoarkkitehtuuri /113/

Käyttöjärjestelmäkerroksen yläpuolella toimivat sovellusten hallintaohjelmisto ja sen yhteyteen toteutettu MHP API. MHP-sovellukset vaativat digi-TV-päätelaitteissa toimivan hallintaohjelmiston, josta käytetään myös nimitystä middleware. Tämä hallintaohjelmisto yhdessä laitteistoon ohjelmoitujen luokkakirjastojen kanssa tarjoavat yhtenäisen rajapinnan vastaanottimen resurssien ja MHP-sovellusten välille /35/, /36/, /113/. Hallintaohjelmisto mahdollistaa datan lataamisen laitteen flash-muistiin, josta sovellukset suoritetaan.

MHP-standardi mahdollistaa sovellusten tietoliikenteeseen useita erilaisia yhteyskäytäntöjä. Sovellukset ja niiden resurssitiedostot ladataan osana broadcast-lähetysten ja paluukanavan yhteyksiin käytetään TCP/IP-protokollaa. Broadcast-dataliikenteessä MHP käyttää DSM-CC-standardin mukaisia datakaruselli- ja oliokaruselliprotokollia. /36/

Koska Java-sovellukset koostuvat useista erillisistä luokkatiedostoista, tulee Java-virtuaalikoneessa olla luokkalataaja, joka vastaa luokkien lataamisesta laitteen muistiin tarpeen mukaan. MHP-standardin mukaan jokaisella digi-TV-päätelaitteella ajettavalla sovelluksella tulee turvallisuussyistä olla oma luokkalataajansa, jotta ne eivät voisi aiheuttaa virhetilanteita toisilleen. Lisäksi ajoympäristön API:lla tulee olla erikseen oma luokkalataajansa.



Tästä syystä päätelaitteessa toimivat MHP-sovellukset eivät suoraan voi kommunikoida keskenään vaan niiden on pyydettävä siihen hallintaohjelmiston lupa. /107/

#### 4.4 MHP-sovellusten hallinta

Jokainen lähetys, joka sisältää MHP-sovelluksen, sisältää myös sovellusinformaatiotaulun (AIT - Application Information Table), johon on kerätty lähetyksen sisältämien MHP-sovellusten tiedot. Näitä sovelluksista tarvittavia tietoja ovat sovelluksen tunnusluku, nimi, sovelluksen ajamiseen tarvittavien tiedostojen sijainti ja sovelluksen käynnistykseen tarvittavat parametrit. /36/

Sovelluksen tunnusluvun tulee olla yksilöllinen ja sen avulla MHP-vastaanottimen hallintaohjelmisto tekee tarvitsemansa viittaukset sovellukseen. Tunnusluku koostuu 32 bitin organisaatiokoodista ja 16 bitin sovelluskoodista. Jokaisella MHP-sovelluksella valmistavalta organisaatiolla on oma organisaatiotunnuksensa ja sen lisäksi organisaation tulee antaa jokaiselle valmistamalleen sovellukselle sovelluskoodi. /35/, /36/

MHP-vastaanottimessa toimiva hallintaohjelmisto käyttää AIT-taulukkoa eräänlaisena sovelluksen hallintaprotokollana. AIT-taulukkoon merkitään tieto mistä kohtaa kanavanipua tietty sovellus ja sen tiedostot löytyvät. Sovellus käynnistetään ja pysäytetään merkitsemällä oikea koodi AIT-taulukkoon. Tämän koodin avulla laitteelle voidaan kertoa annetaanko käyttäjälle oikeudet käynnistää sovellus vai käynnistääkö laite sovelluksen automaattisesti. Tämä antaa palveluntarjoajalle mahdollisuuden ajaa aikasidonnaiset sovellukset oikea-aikaisesti. /79/

Kun käyttäjä vaihtaa katseltavaa kanavaa, vastaanottimen muisti tyhjenee sovelluksista, joita ei ole listattu uuden kanavan AIT-taulussa. Tämä tarkoittaa, että digi-tv-vastaanottimessa voidaan suorittaa usean eri kanavan taustoilla toimivaa MHP-sovellusta vain kaikkien tv-kanavien lähettäjiä yhteisellä sopimuksella. Tämä pätee myös paluukanavan kautta ladattaviin sovelluksiin ja mahdolliselle kiintolevyllä tallennettaviin sovelluksiin. /35/, /79/

#### 4.5 Sovellusten ohjelmointikielet

##### 4.5.1 DVB-Java

Java on ollut luonnollinen valinta digitaalisen television ohjelmointikieleksi, sillä se on erityisesti suunniteltu tietoliikennepäristöjä ajatellen. Java on kehittynyt Internet-maailmassa ja alusta lähtien sen kehityksessä on kiinnitetty erityistä huomiota Internetissä käytössä olevien protokollien tukemiseen. Lisäksi Java on tietoturvallisuudeltaan hyvä ratkaisu, sillä siinä ei ole suoria muistipaikkakutsuja vaan sovellukset toimivat omissa "hiekkalaatikossaan", Sun Microsystemsin kehittämässä ja ylläpitämässä Java-ajoympäristössä. /100/, /102/

Java on hyvin pitkälti laitteistoriippumaton, eli MHP-standardointityöryhmä kykeni valitsemaan sen ohjelmointikieleksi suosimatta näin ketään laitteisto- tai käyttöjärjestelmävalmistajaa. Negatiivinen puoli Javan valinnassa on, että sen tehokkuus erilaisissa kuluttajaelektronikan laitteissa ei ole aivan suoraviivaista. Vaaditaan melko paljon Java-kielen tuntemusta, jotta sillä tehdyt sovellukset toimivat tehokkaasti sulautetuissa järjestelmissä /102/.

Java-ajoympäristö koostuu Java virtuaalikoneesta ja tietyistä funktiokirjastoista eli luokkapaketeista. Nämä kirjastot sisältävät Java-standardin 1.1 mukaiset funktiot eli ohjelmointialustan, jonka päälle sovellukset ohjelmoidaan. Yleisin yhteensopiva JDK (Java Development Kit) on versio 1.1.8. MHP-spesifikaation tukemien java-luokkatiedostojen versionumerot ovat välillä 45.3 – 45.65535 /35/, /36/.

Tällä hetkellä, ennen kuin MHP-päätelaitteisiin tulee sisäänrakennettuja selaimia, Java-kieli on ainoa tapa tehdä MHP-sovelluksia. On jonkin verran epävarmaa tuleeko puhtaita HTML-sovelluksia koskaan näkymään digitaalisessa televisiossa /79/. Kuitenkin tulevaisuutta silmälläpitäen on syytä tarkastella myös DVB-HTML-tekniikkaa.

#### 4.5.2 DVB-HTML

DVB:n eri standardit HTML-kielen ja Java-kielen käyttämisestä sovellusten ohjelmoinnissa eivät ole toistensa kilpailijoita. Siinä missä DVB-Java on tarkoitettu monipuolisempien interaktiivisten sovellusten tekemiseen, on DVB-HTML tarkoitettu lähinnä staattisia tekstipohjaisia sovelluksia varten /35/.

HTML-kielellä tehdyt MHP-sovellukset eivät ole mahdollisia nykyisissä, MHP 1.0-version mukaisissa päätelaitteissa, mutta tulevan version 1.1 mukaiset päätelaitteet tukevat niitä. Markkinoilla on tällä hetkellä joitakin digi-tv:lle tehtyjä selainohjelmia, jotka tukevat HTML-sisältöä. Nämä eivät varsinaisesti ole DVB-HTML-standardin mukaisia, vaan lähinnä DVB-Javalla tehtyjä ohjelmia, jotka tulkaavat HTML-sisältöä. Tällainen selaintyyppinen sovellus täytyy nykyisissä laitteissa ladata ennen sisältöä laitteen muistiin, jolloin se häviää sieltä aina kun ohjelma tai päätelaite suljetaan. /112/

DVB-HTML –standardin ohjelmointikieleksi on tarkoitettu lähinnä XHTML. Kieli on XML-standardin mukainen versio HTML-kielestä ja soveltuu sitä paremmin esimerkiksi monikanavajulkaisemiseen /22/. DVB-HTML-standardi tukee XHTML-kielen lisäksi CSS-tyylitiedostojen versiota 2.0, ECMAScript-kielen ensimmäistä versiota, joka vastaa JavaScriptin versiota 1.3 ja DOM-dokumenttimallin versiota 2.0 /36/, /114/.

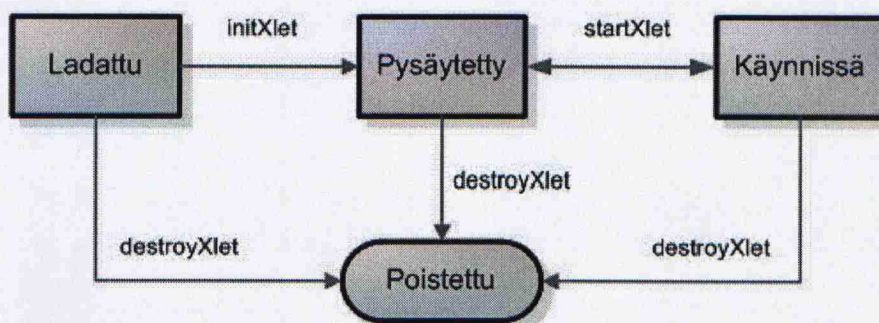
Yksi DVB-HTML-standardin yksityiskohdista on laukaisimien käyttö. Standardin mukaisesti datan lähetysvirtaan voidaan laittaa laukaisimia, joiden avulla sisältö vaihtuu automaattisesti televisioruudulla ilman, että katsoja klikkaa erikseen sivulla olevaa linkkiä /79/. Laukaisimien avulla on mahdollista toteuttaa sovelluksia, joiden sisältö päivittyy automaattisesti tietyn ohjelman edetessä.



## 4.6 Xlet-rajapinta

MHP-sovellukset eivät ole itsenäisiä sovelluksia vaan tietoturvasyistä päätelaite kontrolloi niiden toimintaa erityisen *Xlet*-rajapinnan avulla. MHP-päätelaitteessa toimiva hallintaohjelmisto on vastuussa kaikkien sovellusten käynnistyksestä, pysäyttamisestä ja resurssien jakamisesta eri sovellusten välillä. Mikäli lähetysvirrasta on ladattavissa MHP-sovellus, on tästä kerrottu lähetysvirran sovellusinformaatiotaulussa. Taulussa on kerrottu hallintaohjelmistolle sovelluksen *Xlet*-luokka. Kyseessä on luokka, joka toteuttaa *javax.tv.xlet.Xlet*-rajapinnan ja siten metodit sovelluksen käynnistämiseen ja pysäyttämiseen. Näiden metodien kautta hallintaohjelmisto kontrolloi sovellusta. /35/, /31/

*Xlet*-rajapinta mahdollistaa myös sovelluksen jatkamisen pysäyttämisen jälkeen samasta tilasta. Tästä ominaisuudesta on hyötyä televisioympäristössä, jossa katsoja voi haluta välillä palata katsomaan televisiokuvaa kokonaisuudessaan ja palata tämän jälkeen uudelleen sovellukseen. MHP-sovellus voi olla neljässä tilassa: 1. Ladattu lähetysvirrasta 2. Pysäytetty 3. Käynnistetty 4. Tuhottu / poistettu /35/, /79/. Tilojen väliset suhteet on esitetty seuraavassa kuvassa (Kuva 7). Kuvassa suhteisiin on merkitty uuteen tilaan vievät rajapinnan metodit *initXlet*, *startXlet* ja *destroyXlet*.



**Kuva 7.** *Xlet*-ilmentymän tilat MHP päätelaitteessa /35/

Useimmiten MHP-päätelaitteeseen on toteutettu hallintaohjelmistolle oma käyttöliittymä, josta käyttäjä voi valita käynnistettäväksi tietyn sovelluksen. Sovellus voidaan myös määritellä ladattavaksi tietyn kaukosäätimen komennon jälkeen, kuten Suomen digi-tv:n supertekstitelevisio tapauksessa /95/.

Kun sovellus ladataan, *Xlet*-rajapinnan toteuttavasta luokasta luodaan ilmentymä kutsuamalla sen muodostajaa eli konstruktoria. Mikäli muuta konstruktoria ei ole määritelty, kutsutaan oletuskonstruktoria. Sovellusta käynnistettäessä hallintaohjelmisto suorittaa *initXlet*-metodin, jolloin sovellukselle annetaan parametrina viittaus *Xlet*-kontekstioliioon ja sovellus alustetaan. Kun alustus on valmis, suoritetaan *startXlet*-metodi, jolloin sovellukseen ohjelmoidut toiminnot toteutetaan. *Xlet* tuhoetaan *destroyXlet*-metodilla, jolloin päätelait-

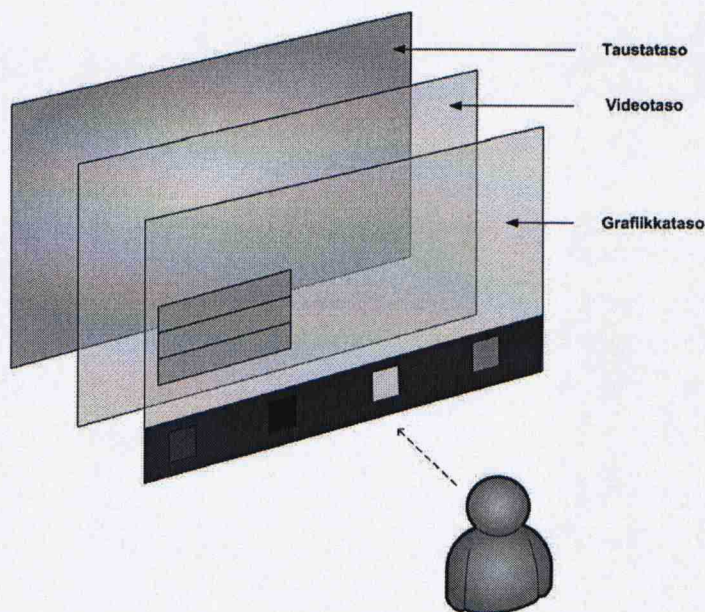
teen muistista hävitetään viittaus luotuun *Xlet*-olioon. *Xlet*-kontekstioliion avulla sovellus voi pyytää ajoympäristökohtaista informaatiota ja kommunikoida hallintaohjelmistolle muutokset omassa tilassaan.

#### 4.7 MHP-sovellusten graafiset käyttöliittymät

MHP-standardin grafiikkamallin ymmärtäminen on MHP-sovellusten laatimisen kannalta tärkeää, sillä sovellukset ovat useimmiten graafisia ja siten grafiikkamalli ohjaa voimakkaasti tapaa, jolla ne rakentuvat. Grafiikan piirtämiseen käytetään Java Abstract Windowing Toolkit API:a ja HAVi spesifikaation GUI API:a. AWT API:n laajentaminen toisella grafiikkakirjastolla on ollut välttämätöntä johtuen television erilaisesta tekniikasta tietokoneen näyttöön verrattuna [30], [31].

Digi-tv-ympäristö eroaa grafiikan käsittelyssä PC-ympäristöstä siinä, että ensiksi mainitun ympäristön päätelaitteissa grafiikan hahmonnus tapahtuu siihen tarkoitukseen erityisesti valmistetulla mikropiirillä. PC-sovelluksissa grafiikan käsittely tapahtuu puolestaan useimmiten ohjelmallisesti, mikä tekee eri toimintojen suorittamisen helpoksi. Digi-tv-päätelaitteiden hintavaatimukset aiheuttavat kuitenkin ainakin toistaiseksi sen, että laitteiden prosessoriteho ei riitä grafiikan ohjelmalliseen käsittelyyn.

Digitaalisessa televisiossa näytön informaatio jakautuu kerrosmallin mukaisesti. Näyttö muodostuu tyypillisesti kolmesta kerroksesta, jotka etumaisesta takimmaiseen ovat grafiikkataso, JMF-videotaso ja taustataso (kuva 8).



**Kuva 8.** Digi-tv:n näytön rakenne MHP-sovelluksissa [68], [95]

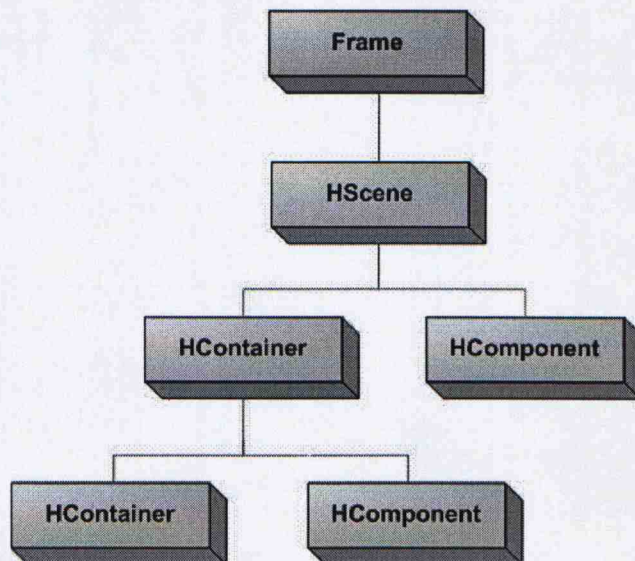


Päätelaitteella suoritettava MHP-sovellus piirtää grafiikkaa etumaiseen tasoon ja voi tehdä JMF API:n avulla videotason videolle tiettyjä operaatioita, kuten kuvan skaalausta ja äänen säätämistä. TV-kuva voi olla sovelluksen taustalla täysikokoisena tai ¼ kokoon pienennettynä. Muiden kuvakokojen tuki riippuu päätelaitevalmistajasta.

Grafiikkatasoja voi olla useampia kuin yksi mikäli halutaan esimerkiksi käyttää tekstitykselle omaa grafiikkatasoa ja valikoille omaansa. Grafiikkatasojen määrä riippuu päätelaitteesta ja suoritettavasta MHP-ohjelmasta. Taustataso voi näyttää tiettyä väriä tai yhtä pysäytettyä MPEG-I-kuvaa, joka voi toimia taustakuvana tarvittaessa. Grafiikkatasoja ohjaa MHP-mallissa *org.havi.ui.HScreen*-luokka, josta päätelaite luo käynnistettäessä ilmentymän jokaiselle siihen kytketylle näytölle. Tyypillisesti päätelaitteeseen on kytkettyinä yksi televisio, joten *HScreen*-ilmentymä luodaan tätä televisiota varten.

Koska digi-tv-päätelaitteet on voitava toteuttaa suppeillakin muistiresursseilla ja pienellä prosessoriteholla, MHP-standardi ei määrittele AWT-käyttöliittymän ikkunamanagerin kaltaista tapaa eri näkymien hallitsemiseksi. Sen sijaan standardissa käytetään HAVi-spesifikaation *org.havi.ui.HScene*-luokkaa näkymän alustana. Sovelluksella voi olla vain yksi *HScene*-ilmentymä, joka vastaa *java.awt.Frame*-luokkaa. Ilmentymään lisätään *org.havi.ui.HComponent*- ja *org.havi.ui.HContainer*-olioita, jotka muodostavat varsinaisen käyttöliittymän (Kuva 9). Ilmentymä pyydetään erilliseltä *org.havi.ui.HSceneFactory*-luokalta, jonka tehtävä on tarjota päätelaitteen ja television ominaisuuksiin mahdollisimman sopiva *HScene*. *HScene*-ilmentymän kautta sovelluksella on mahdollisuus muuttaa televisioon liittyviä graafisia ominaisuuksia kuten kuvasuhdetta ja grafiikkatason läpinäkyvyyttä.

/79/



**Kuva 9.** Käyttöliittymäkomponenttien välinen hierarkia. /79/

Merkittävä ero televisiografiikan ja tietokoneografiikan välillä on, että jälkimmäisessä käytetään RGB-väriavaruutta kun taas televisiossa puolestaan käytetään YUV-avaruutta. Päätelaitteen MHP-alusta huolehtii pääosin värien muutoksista, mutta sovellusohjelmoijan on huolehdittava elementtien läpinäkyvyyden kontrolloimisesta. MHP mahdollistaa grafiikka-erroksen värien läpinäkyvyyden, jolloin videokerroksen video voidaan näyttää grafiikka-elementtien läpi. Käytännössä tämä tapahtuu määrittämällä värille RGB-arvojen (arvot välillä 0-255) lisäksi läpinäkyvyyden alpha-arvo prosenttilukuna. /95/

Graafisiksi komponenteiksi HAVi-kirjasto tarjoaa joukon erilaisia elementtejä. Näitä ovat erilaiset painikkeet, ikonit, rullattavat listat, tekstilaatikot ja tekstin syöttökentät. Elementtien ulkoasun muuttamiseksi MHP käyttää *org.havi.ui.HLook*-rajapintaa ja sen aliluokkia, jotka mahdollistavat graafisten Java-sovellusten käyttämän *paint*-metodin korvaamisen sovelluksessa tietyllä ulkoasulla. *HLook*-ilmentymät ovat käytettävissä vain *org.havi.ui.HVisible*-luokalle, jonka sisään MHP-sovelluksen komponentit asetetaan. /30/, /35/

Grafiikkatason kuva-ala on 720x576 pikseliä TV-kuvan kuvasuhteesta riippumatta. Käytettävä kuva-ala venytetään vaakasuunnassa TV-lähetyssignaalin kuvasuhteen mukaisesti vastaamaan TV-kuvan leveyttä. Televisiossa pikselit eivät ole neliön muotoisia. Grafiikkatason kuvasuhde on 5:4, mutta se venytetään tilanteesta riippuen 4:3 tai 16:9-kuvasuhteen kuvalähetteen päälle. Tällöin grafiikkatason kuvasuhde on 4:3-tilassa 16:15 ja 16:9-tilassa 64:45. Esimerkiksi tietokoneympäristössä käytetyt kuvat täytyy erikseen skaalata, jotta ne näkyisivät televisiossa oikeassa suhteessa. /95/

## 4.8 Syötteiden käsittely

MHP-standardi mahdollistaa käyttäjän syötteiden käsittelyn neljällä eri tavalla. Nämä ovat seuraavat /36/:

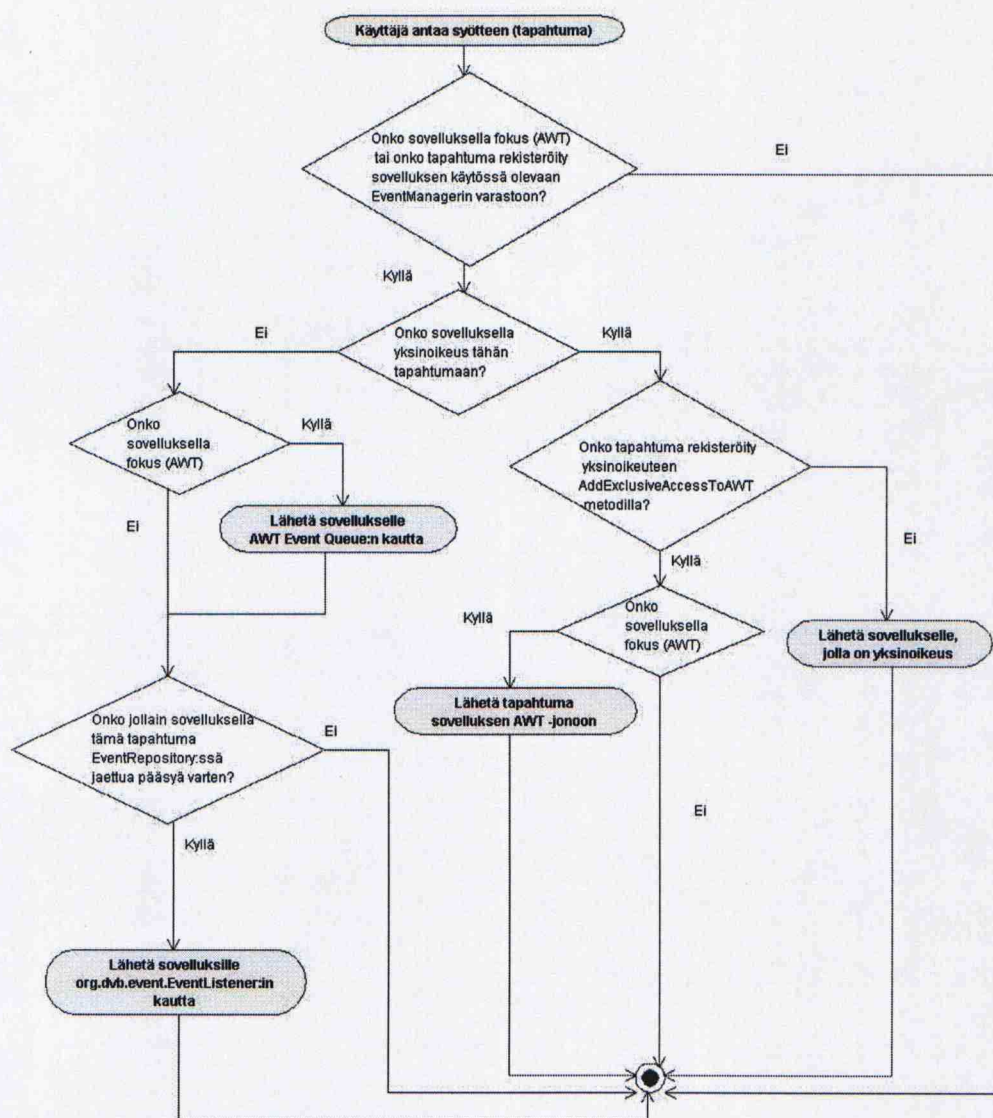
1. Standardien Java AWT-tapahtumankuuntelijoiden avulla
2. Kuten edellisessä, mutta sallien tietyn sovelluksen yksinoikeuden joihinkin tapahtumiin.
3. *org.dvb.event*-API:n avulla.
4. Kuten edellisessä, mutta sallien tietyn sovelluksen yksinoikeuden joihinkin tapahtumiin.

Syy siihen miksi Java AWT:n tapahtumakuuntelumallille on tarvittu vaihtoehtoinen malli on, että AWT tarvitsee koko ajan jonkin käynnissä olevan komponentin, jolla on sovellusympäristön fokus ja joka kuuntelee tapahtumia. Tämä malli vie digi-tv-päätelaitteessa monessa tapauksessa liikaa kapasiteettia muulta prosessoinnilta ja siksi vaihtoehdoksi on kehitetty oma *org.dvb.event*-paketti ja sen *UserEventListener*-luokka /36/. Tämän mallin mukaisesti vastaanotetut tapahtumat eivät siis ole AWT-tapahtumia, vaan luokan *org.dvb.event.UserEvent*-ilmentymiä.



Jotta MHP-sovellus kykenee käsittelemään käyttäjän syötteitä, tulee siinä olla luotuna syötteiden varasto eli *org.dvb.event.UserEventRepository*-olio, johon määritellään kaikki ne syöteryhmät, joita kyseisen sovelluksen tulee olla valmis vastaanottamaan. Pääsy syötteisiin pyydetään erilliseltä *org.dvb.event.EventManager*-oliolta, joka on singleton-tyyppinen olio ja siten yhteinen kaikille päätelaitteissa toimiville sovelluksille.

Sovellukselle voidaan antaa yksinoikeus tiettyihin tapahtumiin, kuten esimerkiksi salasanojen lukemiseen. Tällöin *EventManager*-oliota pyydetään antamaan sovellukselle yksinoikeus tiettyssä *UserEventRepository*-luokassa määriteltyihin tapahtumiin. *EventManager*-olion vastuulla on päättää voidaananko pyydettyihin tapahtumiin antaa yksinoikeus. Tapahtumankuuntelumallin selventämiseksi mallista on alla esitetty UML-kuvauskielen mukainen aktiviteettikaavio (Kuva 10).



**Kuva 10.** MHP:n tapahtumankuuntelumalli /35/

MHP-standardissa on määritelty käyttäjän syötteille kolme laitetta: kaukosäädin, näppäimistö ja hiiri /36/. Laittevalmistajille pakollisia syötelaitteita on vain kaukosäädin /85/. MHP-standardin kaikki sallitut painikkeet on määritelty luokassa *org.havi.ui.event.HRcEvent*.

*UserEvent*-luokan avulla on mahdollista toteuttaa muitakin syötteitä kuin kaukosäätimen tai näppäimistön komentoja. Esimerkiksi viivakoodin lukulaitteen syötteiden käsittely olisi mahdollista toteuttamalla MHP-sovellukseen *UserEvent*-luokalle aliluokka, jossa on määritelty viivakoodin merkkien syötteiden numerot, syötteiden pituus ja luettava lukuavaruus. Tällöin päätelaite vastaanottaa koodin ikään kuin kaukosäätimen komentoina, mutta koo-



din prosessointi tapahtuu sovellukseen ohjelmoidussa aliluokassa. Toinen vaihtoehto on toteuttaa koodin lukulaitteen syötteet näppäimistön komentoina. Tällöin määriteltäisiin *UserEventRepository*:lle aliluokka, jossa toiminnallisuus on määritelty siten, että sovellus osaa odottaa tietyn pituisia tai tietyistä kirjaimista koostuvia merkkijonoja.

## 4.9 MHP-laitteiden Internet-yhteydet

DVB-standardin mukaisesti MHP-sovellukset ja niiden resurssitiedostot ladataan päätelaitteeseen osana digitaalisen television lähetysvirtaa tai Internet Access-profiilin käyttöönoton jälkeen myös käyttäen paluukanavaa. Nykyisessä, vuorovaikutteisessa profiilissa, paluukanavaa käytetään vain sovelluksista lähetettävän informaation välittämiseksi palveluntarjoajalle. Sovelluksia ei siis nyt markkinoilla olevissa laitteissa voi ladata Internetistä.

Sekä vuorovaikutteinen, että Internet Access-profiili käyttävät paluukanavan tietoliikenteessä TCP/IP-yhteyttä, joka voidaan muodostaa esimerkiksi PSTN-, ADSL-, tai kaapeli-modeemilla. Myös mobiilia laitetta kuten matkapuhelinta on mahdollista käyttää paluukanavana /99/. Tällä hetkellä markkinoilla olevissa MHP-päätelaitteissa on paluukanavana PSTN-modeemi, mutta niihin on odotettavissa myös muita edellämäinnittuja paluukanavaratkaisuja /34/.

Vuorovaikutteisuutta sisältävät palvelut voivat joko hyödyntää paluukanavaa tai olla käyttämättä sitä. Kuitenkin, mikäli puhutaan varsinaisista kommunikaatiosovelluksista, on paluukanavan käyttö välttämätöntä. Kommunikaatiosovelluksilla tarkoitetaan tyypillisimmin chat- ja sähköpostisovelluksia. Myös erilaiset lotto-, kauppapaikka-, pankki- ja vedonlyöntisovellukset vaativat paluukanavan olemassaolon.

Sovelluksen arkkitehtuuritasolla yhteyden muodostamiseen käytetään *java.net*-pakettia. Kyseisessä paketissa kuitenkin yleisesti oletetaan, että alemman tason Internet-yhteys on olemassa kun pakettia käytetään. MHP-vastaanottimessa ei kuitenkaan ole käyttöjärjestelmän puolelta tapahtuvaa automaattista verkkoonkytkeytymistä, joten tällainen ominaisuus on pitänyt lisätä MHP API:in. Kyseessä on eräänlainen paluukanavan sessionhallinta, joka on määritelty *org.dvb.net.rc*-paketissa. /35/

## 4.10 MHP ja XML

XML eli eXtensible Markup Language on W3C-konsortion kehittämä standardi dokumenttien metatiedon erottamiseksi sisällöstä. Se on metakieli, jonka avulla voidaan määritellä erilaisia rakenteellisia merkkauškieliä. XML:n avulla sisältöä kuvataan elementeillä. Elementit ja niiden attribuutit erotellaan kulmasuluilla sisällöstä. Jotta elementtien ja dokumentin rakenteen oikeellisuudesta voitaisiin varmistua, on XML-dokumentin mukana mahdollista toimittaa XML-skeema, eli säännöt sisällön kuvaamiselle. Yksi tällainen paljon käytetty skeemakieli on DTD, eli Document Type Definition. /86/



XML on otettu viestintäteollisuudessa laajasti käyttöön, sillä se mahdollistaa saman sisällön julkaisemisen erilaisissa julkaisukanavissa ja päätelaitteissa ilman että sisältöä tarvitsee joka kerta tuottaa uudelleen /81/. XML-tuki onkin ehdoton vaatimus tämän päivän julkaisujärjestelmille. Eri päätelaitteilla on kuitenkin erilaiset graafiset ja ohjelmistokohtaiset ominaisuudet, minkä takia jokainen päätelaite vaatii sille ominaiset tyylitiedostot, jotta sisältö saadaan esitettyä käyttäjälle.

XML soveltuu sisällönkuvauskieleksi varsin hyvin myös MHP-ympäristöön. Jotta XML-tiedostot voidaan siirtää MHP-vastaanottimelle, täytyy ne lähettää joko lähetysvirran oliokarusellilla tai käyttämällä kaksisuuntaista paluukanavayhteyttä. Vastaanottimen tulee tällöin ladata sovelluksen tarvitsemat XML-dokumentit ja erottaa (parsia) sisältö metatiedosta. XML:n parsimiseen Java-kielessä on yleisesti käytössä kaksi hieman toisistaan poikkeavaa ratkaisua: Simple API for XML eli SAX ja Document Object Model eli DOM.

SAX on tapahtumapohjainen parseri, joka lukee sille annetun XML-dokumentin alusta loppuun ja ilmoittaa parseria ajavalle sovellukselle kaikista XML-syntaksin mukaisista tapahtumista, eli XML-elementin alkamisesta, elementin sisällöstä ja elementin loppumisesta. DOM puolestaan lukee XML-tiedoston ja rakentaa siitä dokumenttipuun, eli säilyttää muistissa dokumentin rakenteen ja tiedot elementtien välisistä suhteista puumaisena tiedostona. DOM:in avulla elementtejä voidaan lisätä, muokata ja poistaa dokumenttipuusta. /1/, /81/

SAX:in kaltaiset ratkaisut soveltuvat usein MHP-sovelluksiin parhaiten, kun dokumentin sisältö halutaan vain nopeasti luettua ja näytettyä. SAX on toteutuksena paljon kevyempi kuin DOM, joka käyttää dokumenttipuun rakentamiseen ja muistissa pitämiseen paljon muistiresursseja. SAX-parserista on olemassa vielä huomattavasti kevennettyjä ratkaisuja, mikäli muistiresurssien käyttö halutaan minimoida. /33/, /112/

#### 4.11 Sovellusten tallentaminen päätelaitteelle

MHP-standardin versio 1.1 mahdollistaa tiettyjen sovellusten tallentamisen päätelaitteen muistiin niin, ettei niitä tarvitse joka käynnistyskerralla ladata uudelleen. Tämä vaatii luonnollisesti, että päätelaitteella on käytettävissä kovalevy tai muu massamuisti, jolle sovelluksia voidaan tallentaa. Tallentaminen tapahtuu siten, että palveluntarjoaja lisää AIT-tauluun tiedon, joka kertoo päätelaitteelle tietyn sovelluksen tallentamisesta. Tällöin sovellus jää vastaanottimen muistiin, mikäli käyttäjä on antanut tähän oikeuden. AIT-taulun tietojen lisäksi lähetykseen lisätään myös sovelluksen kuvaustiedosto, joka kertoo mikä lähetyksen tiedostot kuuluvat tallennettavaan sovellukseen. /35/

On todennäköistä, että MHP-standardin tulevien versioiden mukaisissa laitteissa tiettyjä sovelluksia ladataan pysyvästi laitteen muistiin niiden käynnistämisen nopeuttamiseksi. Ensimmäiset tallennettavat sovellukset ovat luultavimmin selainohjelmia, joiden avulla esimerkiksi superteksti-tv:n sisältöjen katselu nopeutuu. Tällaiset selaimet voidaan nähdä eräänlaisina sovelluskehysinä HTML- ja XML-sisällöille. Kun sovelluskehysen tallentaa

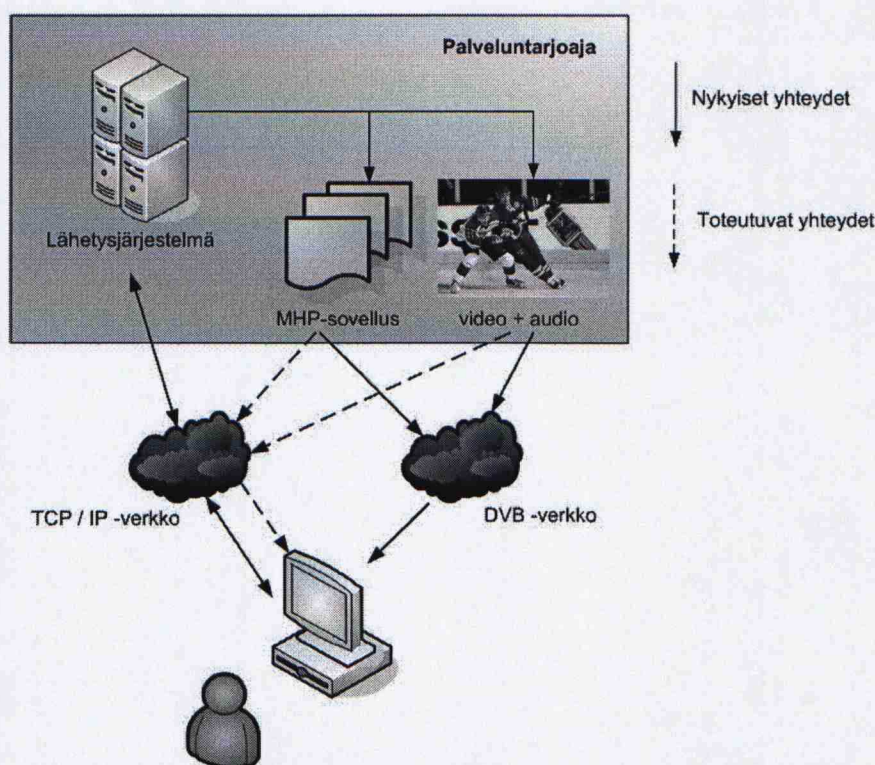


laitteen muistiin, tarvitsee lähetyksestä ladata vain palvelun sisältö. Tällainen malli todennäköisesti yleistyy MHP-vastaanottimien tallennuskapasiteetin kasvaessa.

#### 4.12 MHP:n tulevaisuudennäkymät

Kaiken kaikkiaan MHP-teknologia on vielä kehityksensä alkuvaiheessa. Palveluiden käytettävyyden suurimpina ongelmina ovat sovellusten hitaus ja MHP-päätelaitteiden epävarma MHP-yhteensopivuus /34/. Nämä ongelmat johtuvat DVB-MHP-standardin monimutkaisuudesta ja oliokarusellin hitaudesta käsiteltäessä sovelluksia joissa on useita suurikokoisia tiedostoja. Lisäksi nykyisten päätelaitteiden prosessointiteho on suhteellisen pieni ja kyky välimuistittaa sovellusten tiedostoja on puutteellinen. MHP-päätelaitteet kehittyvät kuitenkin ajan kuluessa suorituskyvyltään tehokkaammiksi.

Laitteiden tehokkaamman prosessoinnin ja oliokarusellista ladattavien tiedostojen välimuistittamisen myötä MHP-sovellusten käytettävyys todennäköisesti paranee. Lisäksi paluukanavassa siirrytään käyttämään laajakaistayhteyksiä, jolloin sovellusten ja jopa videon lataaminen Internetistä mahdollistuu. Tällaisessa ympäristössä päätelaitteen ja sovelluspalvelimen välisiä TCP/IP-yhteyksiä voidaan käyttää nykyisen oliokarusellin sijaan (Kuva 11). /36/



**Kuva 11.** Digitaalisen television palveluiden tiedonkulku ja yhteydet

MHP:n seuraavat versiot tulevat näillä näkymin sisältämään ohjelmointirajapinnat MHP-päätelaitteen kovalevyn hallintaan ja nykyistä kehittyneempään kommunikaatioon lähiverkossa toimivien laitteiden kanssa OSGI-spesifikaation mallin mukaisesti /36/, /88/. Tämän kehityksen myötä digi-tv:llä on mahdollisuus muuttua perinteisestä televisiosta monipuoliseksi kodin viihdekeskukseksi, jossa katsojille tarjotaan vuorovaikutteisia video-on-demand-konsepteja hyödyntäen sekä laajakaistayhteyksiä että DVB-lähetyksiä. Kapasiteettin salliessa lähetyksiä voidaan jopa lähettää HDTV-resoluutiolla /93/.

Tehokkaammat videonpakkausstandardit kuten MPEG-4 part 10 otetaan todennäköisesti käyttöön myös MHP-ympäristössä, jolloin voidaan tarjota korkealaatuisia palveluita nykyistä pienemmällä tietoliikennekapasiteetilla. MHP-laitteista tulee tehokkaampia ja niiden grafiikkaominaisuudet paranevat. Tällaisissa laitteissa television katselu ja tietokonepelaaminen lähenevät toisiaan.



## 5 PRINTACCESS MHP-YMPÄRISTÖSSÄ

### 5.1 Yleistä

Diplomityössä on tähän mennessä käyty läpi DVB-MHP-teknologiaa, jotta sen ominaisuudet ja erityispiirteet kävisivät selviksi. Tästä eteenpäin työssä keskitytään sovittamaan PrintAccess-sovelluksia tähän teknologiaan ja löytämään ne vaatimukset, joita nämä sovellukset asettavat MHP-ympäristölle. Näitä vaatimuksia käyttäen suunnitellaan ja toteutetaan lopuksi sovelluskehitys, jonka avulla PrintAccess-sovelluskehitys helpottuu.

PrintAccessin käyttöliittymänä toimii yleisesti ottaen se piste, jonka kautta on pääsy varsinaiseen informaatioon. Tämä piste voi olla painotuotteeseen painettu viivakoodi tai jokin muu kuvainformaatiosta tulkittavissa oleva koodaustapa. Perinteisesti tällaista käyttöliittymää on käytetty esimerkiksi sanomalehteen painettujen televisiokanavien ohjelmätietojen lukemiseen viivakoodista videonauhuria varten. Tulkittaessa koodia painotuotteesta, voidaan koodi siirtää lukulaitteella päätelaitteelle prosessoitavaksi tai koodi voidaan tulkita jo lukulaitteessa.

Nykyisin käytössä olevia esimerkkejä automaattisista hybridimenetelmistä ovat viivakoodien ja www-osoitteiden koneellinen luku painotuotteesta ja niihin perustuva kytkeytyminen sähköiseen palveluun. Tällaiset sovellukset eivät kuitenkaan ole saavuttaneet erityisen suurta suosiota, minkä on tulkittu johtuvan siitä, että sovellusten käyttö on edellyttänyt erillistä koodin lukulaitetta, jota käyttäjien on hankala erikseen kantaa mukanaan /67/. Kameratelevisioiden matkapuhelinten myötä tilanne kuitenkin muuttuu sikäli, että yhä useammalla kuluttajalla on koko ajan mukanaan laite, jolla koodien lukeminen painotuotteesta on mahdollista.

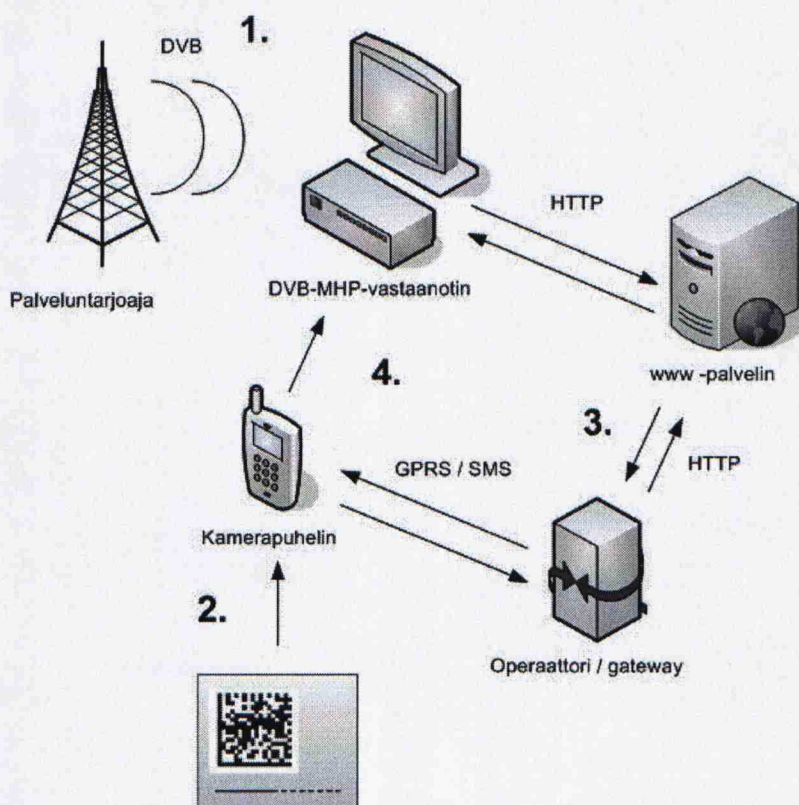
Digitaalisen television MHP-ohjelmointiympäristö on houkutteleva kohde PrintAccess-koodien käyttöön, sillä koodi mahdollistaa esimerkiksi laskutettavan pääsyn olemassaoleviin DVB-MHP-sovelluksiin tai niiden lisäinformaatioon. MHP-ympäristö on kiinnostava PrintAccess-konsepteille myös sikäli, että MHP-ominaisuuksilla varustettu digi-tv-päätelaite voi tulevaisuudessa hyvinkin olla joka kodin keskeinen multimedialaite. Näiden lisäksi digitaalisessa televisiossa on olemassa olevat sisällöntuotantokanavat, joiden sisältöä ihmiset ovat tottuneet käyttämään. Päätelaite mahdollistaa tulevaisuudessa myös Internetin palvelujen kytkemisen televisiosisältöihin ja sen myötä aivan uudenlaisten ohjelmakonseptien luomisen.

### 5.2 PrintAccess ja MHP-vastaanotin

Kun painotuotteessa oleva koodi on saatu luettua jollakin keinolla, sen informaatio tulee siirtää MHP-päätelaitteelle, jotta MHP-sovellusten ohjaus koodin avulla olisi mahdollista. MHP-päätelaitteella tulee näin olla sovellus, joka osaa ottaa vastaan sille lähetetyn koodin joko tulkitussa muodossa tai kuvamuodossa. Mikäli koodi lähetetään kuvamuodossa sovellukselle, tulee sovelluksen osata lukea ja avata koodi kuvasta. Muitakin vaihtoehtoja koo-

din tulkitsemiselle on. Koodi voidaan tulkitä lukulaitteessa tai se voidaan lähettää lukulaitteesta kolmannelle osapuolelle (operaattori), joka tulkitsee koodin ja lähettää sitä vastaavan informaation paluuviestinä joko takaisin lukulaitteelle tai suoraan MHP-päätelaitteelle. Kuvion mukainen tilanne on esitetty alla olevassa kuvassa (Kuva 12). Kuvan numeroinnit ovat seuraavat:

1. MHP-ohjelma ladataan päätelaitteelle osana DVB-lähetystä.
2. PrintAccess-koodi luetaan käyttäen lukulaitetta (esim. kameralla varustettu matkapuhelin)
3. Mahdollinen tilanne: koodi varmennetaan ja sitä vastaavat toimintaohjeet haetaan WWW-palvelimelta käyttäen paluukanavayhteyttä.  
(Tätä yhteyttä käyttämällä voidaan ladata myös MHP-ohjelma mikäli päätelaite tukee Internet Access-profiilia).
4. Koodi siirretään matkapuhelimelta MHP-päätelaitteelle käyttäen kaukosäädinrajapintaa tai paluukanavan tiedonsiirtoprotokollaa.



**Kuva 12.** PrintAccess-informaation tiedonkulku MHP-vastaanotossa



### 5.3 PrintAccess-koodit

PrintAccess-koodien laatimiseen voidaan käyttää kaikkia niitä tekniikoita, joiden avulla painotuotteeseen saadaan erillisillä lukulaitteilla luettavissa olevia koodeja. Koodit voivat olla painotuotteessa esimerkiksi selväkielisenä tekstinä tai painettuna symbologiana kuten viivakoodina. Koodit voivat olla myös piilotettuina kuvaan tai painettuna näkymättömällä, UV- tai IR-valolle herkällä musteella. Tulevaisuudessa tällainen koodipainatus voidaan tehdä esimerkiksi sähköisellä musteella, jolloin sen sisältämää informaatiota voi päivittää vielä painamisen jälkeenkin.

Diplomityön osalta rajoitutaan kuvaamaan tarkemmin näkyviä koodeja. Tällöin ne ovat joko viivakoodia tai selkokieliä avainsanoja. Jälkimmäisten lukemiseen painotuotteesta voidaan käyttää OCR-teknologiaa (Optical Character Recognition).

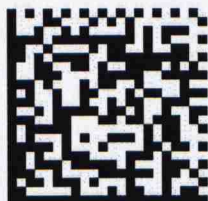
#### 5.3.1 Viivakoodit

Viivakoodia on perinteisesti pidetty melko alkeellisena esimerkkinä hybridimediasta. Näkemyksenä on ollut, että luotettavuudesta huolimatta viivakoodit sopivat visuaalisesti huonosti painoasuun eikä niitä voida selväkielisesti lukea /58/. Kuitenkin viivakooditekniologia tarjoaa yksinkertaisen ja edullisen tavan koodata teksti- ja numeroinformaatiota selkaiseen muotoon että se on luettavissa elektronisilla lukulaitteilla. Tällainen lukulaite voidaan liittää osaksi esimerkiksi kaukosäädintä tai matkapuhelinta. Matkapuhelimeen integroitu kamera voi myös toimia viivakoodin lukijana, jolloin koodi tulkitaan kuvasta ohjelmallisesti /62/.

Viivakoodien tarkoituksena on pääasiassa merkitä tietty elementti yksilöllisellä numero- tai tekstijonolla. Viivakoodia käytetään tyypillisesti yhdistettynä tietokantasovelluksiin siten, että viivakoodin numero on avain jolla tietokannasta haetaan ko. elementin tiedot.

Tällä hetkellä käytössä on useita erilaisia viivakoodisymbologioita, jotka eroavat toisistaan koodaustavan ja merkkivalikoiman suhteen. Yleisimmät viivakoodit ovat lineaarisia, mutta myös kaksiulotteiset koodit ovat yleistymässä. Viimeksimainittujen tarkoituksena on kyetä koodaamaan suurempaa datamäärää kuin lineaarisilla koodeilla. Esimerkiksi DataMatrix-koodilla on mahdollista koodata jopa 2000 alfanumeerista merkkiä yhteen viivakoodisymboliin /55/. Symbolin koko tosin kasvaa siihen sisällytetyn datan suhteessa, jolloin painotuotteen ulkoasu muodostuu useissa tapauksissa rajoittavaksi tekijäksi koodattavan informaation määrälle.

Esimerkki 20x20 DataMatrix-koodista on seuraavassa kuvassa (Kuva 13). Kuvan koodi on DataMatrix ECC200-standardin mukainen ja sisältää tekstisisällön lisäksi virheenkorjausinformaatiota /55/. Kapasiteettinsa ansiosta DataMatrix-koodit ovat yksi hyvä vaihtoehto PrintAccess-sovellusten koodeiksi ja niitä onkin PrintAccess-projektissa päädytty käyttämään ensimmäisen vaiheen sovelluksissa.



**Kuva 13.** *Esimerkki 20x20 DataMatrix ECC200 – viivakoodista. Symboliin on koodattu tekstisisältö: ”TKK Viestintätekniikan laboratorio”*

### 5.3.2 OCR –teknologiat

OCR eli Optical Character Recognition on tekniikka, jonka avulla paperidokumentteja voidaan skannata ja tulkita dokumentin sisältämät kirjaimet digitaaliseen muotoon. OCR toimii samalla tavoin kuin biometriset tunnistuslaitteet, eli fotosensoreilla määritetään tietyn kirjaimen muoto ja sitä verrataan tunnettujen kirjainten tietokantaan. Tunnistetut kirjaimet muutetaan esimerkiksi ASCII –formaattiin, jolloin teksti on käytettävissä erilaisissa soveluksissa. Matkapuhelimiin rakennetut kamerat voivat OCR-tekniikan avulla toimia pienikokoisina skannereina ja niiden avulla voidaan lukea esimerkiksi selkokielisiä URL-osoitteita ja puhelinnumeroita suoraan lehdestä puhelimen muistiin. /68/

Nykyisissä OCR –laitteissa käytetään varsin monimutkaisia algoritmeja ja tekoälyä kirjainten tunnistamiseen mahdollisimman tarkasti. Ohjelmistot eivät ainoastaan tunnista kuvioita vaan osaavat myös tehdä oletuksia sanoista ja kirjainten oikeasta järjestyksestä. Tekniikkaa käytetään esimerkiksi korjaamaan perinteisessä skannauksessa syntyneitä virheitä. Tämä vaatii lukijalaitteelta enemmän prosessointitehoa kuin esimerkiksi viivakoodien lukeminen ja saattaa siksi olla viivakoodeihin verrattuna vähemmän houkutteleva teknologia. /116/

## 5.4 PrintAccess-koodien lukulaitteet

Yhtenä suurimmista vaikeuksista television ja Internetin yhteiskäytössä on tähän saakka pidetty tekstin, eli lähinnä URL-osoitteiden syöttämistä laitteelle. Tarkoitukseen on kehitetty langattomia näppäimistöjä, mutta ne on koettu melko hankalaksi käyttää esimerkiksi sohvalta käsin /58/, /95/. PrintAccess-koodin lukulaite voi koodin lukuominaisuuden avulla toimia interaktiivälineenä, joka mahdollistaa Internetin käytön MHP-päätelaitteella ilman näppäimistöä. Tällöin käyttäjä skannaa URL-osoitteen lehdistä tai kirjoista, jolloin viestinten eri ominaisuudet yhdistyvät toisiaan palvelevilla tavoilla. Skannauksessa voidaan käyttää esimerkiksi mobiiliin laitteeseen integroitua viivakoodin lukijaa tai kameraa. Myös tietokoneeseen yhdistettyä web-kameraa voidaan käyttää painotuotteessa olevien koodien kuvaamiseen ja lukemiseen. Tällöin koodin tulkinta tapahtuu tietokoneella, jolta koodin informaatio siirretään MHP-päätelaitteelle. Tällöin kuitenkin joudutaan tinkimään mobiililaitteen käytettävyydestä.



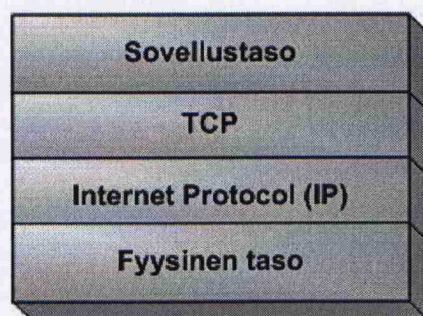
Edellämainittu mobiili laite voi olla esimerkiksi päätelaitteen natiivi kaukosäädin, jossa olisi integroitu kamera tai kamerallinen matkapuhelin. Matkapuhelin tuntuu hyvältä vaihtoehdolta, sillä useissa malleissa on jo nyt integroitu kamera ja nykyisten matkapuhelinten ohjelmistotalustalle on laadittavissa sovelluksia, jotka ohjaavat kodin viihdelaitteita.

## 5.5 MHP-päätteen ja lukulaitteen välinen tietoliikenne

Mikäli lukulaitteelta, kuten matkapuhelimelta, siirretään informaatiota päätelaitteelle lyhyen kantaman tietoliikenneyhteydellä, tulee käytettävän protokollan olla kantamaltaan riittävän suuri ja tehonkulutukseltaan pieni. Tällaisia ovat esimerkiksi tietyt IrDA –infrapunaprotokollan versiot ja Bluetooth-yhteydet. Useissa markkinoilla olevissa matkapuhelimeissa nämä protokollat ovat jo toteutettuina. Tulevaisuudessa myös WLAN-yhteyden käyttö tähän tarkoitukseen on mahdollinen, kun WLAN-teknologia tulee osaksi matkapuhelimia /97/.

Markkinoilla olevissa MHP-laitteissa käytetään infrapunaprotokollaa kaukosäätimen toimintojen suorittamiseen. Muut yhteystavat kuten suurten tiedostojen siirto on mahdollista toteuttaa käyttäen laitteen paluukanavaa. Tällöin paluukanavan teknologia määrää käytettävän fyysisen tason protokollan. Nykyisissä laitteissa tämä on PSTN-modeemi. Mikäli halutaan käyttää Bluetooth- tai WLAN-yhteyksiä, on paluukanava toteutettava näiden teknologioiden avulla. MHP-standardi ei ota kantaa siihen millaisilla fyysisen tason laitteilla kaukosäätimen tai paluukanavan toiminnot toteutetaan. Näiden toteuttaminen riippuu päätelaitteen valmistajasta. On todennäköistä, että MHP-vastaanottimien kehittyessä markkinoille tulee myös näillä tekniikoilla varustettuja laitteita. /99/

Korkean tason yhteysprotokollana käytetään kaukosäätimissä valmistajakohtaista infrapunaprotokollaa ja paluukanavayhteyksissä TCP/IP-protokollaa (Kuva 14). TCP/IP-yhteyksissä tarvitaan fyysisen ja TCP/IP-tasojen lisäksi lisäksi sovellustason protokolla, jotta sovellukset voivat kommunikoida toistensa kanssa. Laitteiden tulee tietää TCP/IP-yhteyksissä toistensa IP-osoitteet, jotta yhteyden muodostaminen onnistuu. /46/



**Kuva 14. TCP/IP-arkkitehtuuri /46/**

## 5.6 Tietoliikenteen fyysinen taso

### 5.6.1 Infrapuna

Kuten edellä todettiin, käytetään markkinoilla olevissa digi-tv-laitteissa kaukosäätimen siirtotienä infrapunaa. Infrapuna on sähkömagneettista säteilyä, jonka taajuus on energiaspektrissä näkyvän valon ja radiotaajuuksien välillä. Infrapunasäteilyn aallonpituus on 780 nm – 1 mm ja taajuus välillä  $3,85 \times 10^{14} - 3 \times 10^{11}$  Hz /3/. Tiedonsiirrossa laitteiden IR-moduulien tulee olla suunnattu toisiinsa esteettömästi, sillä infrapunasäteet eivät läpäise esteitä. Useimmissa infrapunalaitteissa käytetään suunnattua lähetystä ja pientä 15-30 asteen lähetyskartiota, jolloin informaatio saadaan suhteellisen varmasti ja vähäisellä tehonkulutuksella vastaanottajalle /77/. Tehonkulutus on vähäistä, koska lähettävään laitteeseen tarvitsee kytkeä virta vain signaalin lähettämisen ajaksi. Tämän vuoksi tekniikkaa käytetään kaukosäätimissä, sillä sen avulla kaukosäätimen paristot riittävät pitkään. Infrapunatekniikka soveltuu parhaiten yhden huoneen sisällä tapahtuvaan tiedonsiirtoon tai kahden lähekkäin sijaitsevan laitteen väliseen tiedonsiirtoon.

Kaukosäätimien infrapunayhteyksien kantoaaltoa moduloidaan eri tavoin. Mikäli tiedonsiirto toteutettaisiin vain kytkemällä infrapuna-LED päälle ja pois päältä, olisi se hyvin häiriöaltis taustavalolle. Siksi signaali moduloidaan lähettävässä kaukosäätimessä yleisimmin 36 kHz:n taajuudelle. Vastaanottavassa laitteessa kaistanpäästösuodatin poistaa tarpeettomat taajuudet. Jokainen pulssi on itseasiassa joukko infrapunapurskeita kantoaallon taajuudella. Infrapunatunnistin havaitsee fotoelektrisen vaikutuksen, joka muunnetaan jännitteen muutokseksi. /9/

Infrapunatiedonsiirtoon käytetään erilaisia protokollia, joista esimerkkejä ovat RC-5, RECS-80, Sony ja IrDA. Kolme ensimmäistä ovat yleisesti käytössä kulutuselektroniikan kauko-ohjaimissa, kun taas IrDA on laajalti käytetty tilanteissa, joissa siirrettävän tiedon määrä on suurempi kuten esimerkiksi siirrettäessä tietoa tietokoneiden ja mobiililaitteiden välillä.

Eri digi-tv-päätelaittevalmistajien infrapunarajapinnoissa käytetään useimmiten valmistajakohtaisia protokollia. Matkapuhelimissa ja PDA-laitteissa käytettävä infrapunaprotokolla on puolestaan IrDA-standardin mukainen. IrDA-laitteisiin voidaan ladata valmistajakohtainen infrapunaprotokolla valmiina tai ne voidaan opettaa käyttämään tietyn valmistajan valmistajakohtaista infrapunaprotokollaa. Eri vaihtoehdot ohjaimen ja päätelaitteen väliselle infrapunayhteydelle ovat näin ollen:

- Ohjainlaite käyttää natiivisti päätelaitteen valmistajakohtaista infrapunaprotokollaa tai siihen ladataan valmistajakohtainen profiili esimerkiksi valmistajan [www-sivuilta](http://www.sivuili.fi).
- Ohjain opetetetaan käyttämään tietyn päätelaitteen infrapunaprotokollaa.
- Markkinoille tulee digi-tv-päätelaitteita, joissa tuettu infrapunaprotokolla on IrDA.



Ohjaimen opettaminen tapahtuu siihen ladattavalla ohjelmistolla. Tarkoitukseen käytettävät ohjelmistot toimivat yleisesti kahdella eri tavalla. Ohjelmisto voi joko sisältää valmiiksi kulutuselektroniikan laitevalmistajien käyttämät infrapunaprotokollat tai ohjelmistolle voidaan komento kerrallaan syöttää informaatio siitä mikä infrapunasignaali vastaa mitään komentoa. Jälkimmäinen vaihtoehto tapahtuu viemällä oppiva laite ja päätelaitteen kaukosäädin vastakkain siten, että niiden infrapunamodulit ovat samalla kohtaa. Tämän jälkeen oppivalle laitteelle välitetään tieto mikä komento milloinkin on tulossa. Kun kaukosäätimestä sitten painetaan tiettyä komentoa, rekisteröi ohjain komennon infrapunasignaalin ja voi toistaa sen täsmälleen samanlaisena tarvittaessa.

IrDA-protokollan käyttämisessä on myös tietyt erityisvaatimukset. IrDA –standardissa määritellään lähettävän laitteen säteilyn intensiteetin aallonpituudeksi 850-900 nm ja intensiteetiksi 500 mW/sr /77/. IrDA:n kaksi päästandardia ovat IrDA Control ja IrDA Data, joista Controlia käytetään käskyjen ja kontrollisanomien siirtämiseen ja Dataa tiedostojen välittämiseen.

Mobiililaitteita varten on oma IrDA IrMC-spesifikaatio, jossa myös liikennöintitilan virrankulutus on minimoitu /63/. IrMC:n dataa lähettävän IR-ledin ohjausvirta on matala, jolloin myös toimintaetäisyys lyhentyi 20-30 cm:iin. Toisen IrMC-laitteen kanssa viestintäetäisyys on 20 cm ja muun IrDA-laitteen kanssa 30 cm. Tämä tarkoittaa PrintAccessin tapauksessa sitä, että koodit tulee siirtää hyvin läheltä päätelaitetta, mikä saattaa aiheuttaa ongelmia eri konseptien käytettävyydessä. Eri IrDA-protokollat ja niiden ominaisuudet on esitetty alla olevassa taulukossa (Taulukko 4).

**Taulukko 4.** IrDA –protokollat /63/ /77/

|                           | IrDA Control | IrDA Data                | IrDA IrMC                       |
|---------------------------|--------------|--------------------------|---------------------------------|
| <b>viestintätarkoitus</b> | ohjauskäskyt | tiedostot                | mobiililaitteiden kommunikointi |
| <b>siirtonopeus</b>       | 75 kbit/s    | 9,6 kbit/s –<br>4 Mbit/s | 9,6 kbit/s –<br>4 Mbit/s        |
| <b>siirtoetäisyys</b>     | 5 m          | 1 m                      | 20 – 30 cm                      |

### 5.6.2 Bluetooth

Bluetooth on avoin standardi, joka mahdollistaa useiden laitteiden keskinäisen viestinnän noin 10 m etäisyyteen saakka maksiminopeudella 780 kbit/s. Siinä missä infrapunayhteyksissä käytetään IR-alueen alhaisia aallonpituuksia, on Bluetoothissa käytössä korkeat radioaajuudet. Tämä tarkoittaa, että Bluetoothin avulla laitteet voivat olla yhteydessä ilman välitöntä näköyhteyttä. Markkinoilla on digitaalisen television päätelaitteita, joissa on Bluetooth-tuki, mutta näissä ominaisuutta ei ole yhdistetty MHP-alustaan. Tiedon siirto päätelaitteelle tapahtuu olemassa olevissa toteutuksissa OBEX Object Push-tekniikalla, jolla voidaan siirtää esimerkiksi kuvia kahden eri Bluetooth-laitteen välillä /17/. Bluetoothilla

arvioidaan olevan edellytyksiä yleistyä laajemminkin kodin kulutuselektroniikan laitteiden välisessä kommunikaatiossa /108/. Useissa markkinoilla olevissa matkapuhelimissa on Bluetooth-tuki.

Bluetoothin kehittämisen aloitti Ericsson vuonna 1994 ja vuonna 1998 mukaan liittyivät Intel, Toshiba, IBM ja Nokia. Nykyisin Bluetooth –yhteisössä on lukuisia merkittäviä laite- ja ohjelmistovalmistajia mukaanlukien useimmat digi-TV-päätelaitteiden valmistajat. Bluetoothin suunnittelun tärkeimmät lähtökohdat ovat olleet alhainen tehonkulutus ja hinta sekä kyky tukea ad hoc-verkottumista. Bluetooth-tekniikan suunnittelussa on ollut tavoitteena helppokäyttöisyys, nopea tiedonsiirto, multimediatuki ja globaali toimivuus sekä yhteensopivuus. Laitteisiin tarvitaan vain yksi ja massatuotettuna edullinen Bluetooth-siru. Bluetoothin bruttosiirtonopeus (laitteet kommunikoivat samanaikaisesti) on 1 Mbit /s. /20/, /78/

Bluetooth-yhteyteen käytetään 2,4 GHz ISM taajuuskaistaa eli 2400 – 2483,5 MHz taajuutta /53/. ISM-kaista (Industrial, Scientific and Medical band) on avattu matalatehoisille laitteille, jotka voivat toimia ilman Viestintäviraston myöntämää radiolupaa /14/. Kaistalla on paljon häiriöitä, sillä sitä käytetään myös esimerkiksi vauvojen itkuhälyttimissä ja autonovien avaamisessa. Bluetoothissa käytetään nopeaa taajuushyppelyä signaalin sekoittumisen estämiseksi. Tällöin lähettimet vaihtavat taajuutta noin 1600 kertaa sekunnissa, jolloin on hyvin epätodennäköistä että kaksi lähetintä on samalla taajuudella samaan aikaan. Lähettävät paketit pidetään pieninä ja ne lähetetään taajuushyppelyn mukaisesti eri taajuuksilla. Tämä tekee lähetystä ja vastaanottamista hoitavat laitteet melko monimutkaisiksi, sillä molempien laitteiden satunnainen taajuushyppely tulee synkronoida viestinnän mahdollistamiseksi. /20/, /78/

Bluetooth-laitteet tunnistavat toisensa tullessaan kommunikaatioetäisyydelle. Tällöin laitteet muodostavat lähiverkon, jota kutsutaan nimellä Piconet. Kun tällainen lähiverkko muodostuu, laitteet aloittavat samanaikaisen taajuushyppelyn ja siten erottuvat muista samassa tilassa mahdollisesti toimivista lähiverkoista. /20/

Bluetooth sisältää protokollan alempien kerrosten ja sovelluskerroksen väliseen kommunikointiin. Bluetooth voi myös toimia kuljetuskerroksena IP-pohjaiselle liikenteelle jonka päälle sovellus voidaan toteuttaa esimerkiksi sovelluksen ohjelmointikielen tietoliikennerajapinnalla /78/. MHP-alustaan Bluetooth voidaan näin integroida käyttämällä sitä paluukanavan toteuttavana tietoliikenneyhteytenä /99/.

### 5.6.3 WLAN

WLAN –termillä (Wireless Local Area Network) tarkoitetaan joukkoa langattoman lähiverkon muodostamiseen käytettäviä standardeja. Alkuperäinen standardi on IEEE 802.11, joka julkaistiin vuonna 1997. Kyseinen standardi mahdollistaa 2Mbit /s liikenteen 2,4 GHz:n taajuudella. Standardista on julkaistu uudet versiot 802.11a, 802.11b ja 802.11g,



jotka mahdollistavat suuremmat liikennöintinopeudet. Standardien ominaisuudet on koottu alla olevaan taulukkoon (Taulukko 5). /101/, /97/

**Taulukko 5. WLAN-standardin versioiden vertailua**

| Standardi | Maksiminopeus | Taajuus |
|-----------|---------------|---------|
| 802.11    | 2 Mbit/s      | 2,4 GHz |
| 802.11a   | 54 Mbit/s     | 5 GHz   |
| 802.11b   | 11 Mbit/s     | 2,4 GHz |
| 802.11g   | 54 Mbit/s     | 2,4 GHz |

WLAN –tekniikka on alunperin suunniteltu lähinnä pöytätietokoneiden ja sylimikrojen väliseen kommunikaatioon. Laitteet kommunikoivat melko korkeilla maksiminopeuksilla ja laitteiden etäisyydet voivat olla yli 100 metriä. Yhteyksien tiedonsiirtoprotokollana on useimmiten TCP/IP.

Edellä mainituista syistä WLAN-tekniikoilla on melko korkeat tehovaatimukset, mikä vähentää niiden käyttökelpoisuutta mobiililaitteissa. Matkapuhelinvalmistajat ovat kuitenkin viime aikoina kiinnostuneet liittämään WLAN-toiminnallisuuden osaksi puhelimia lähinnä standardin mahdollistamien VoIP (Voice over IP) –puhelujen vuoksi /71/. Tällaiset puhelut ovat erittäin edullisia käyttäjälle, eivätkä välttämättä tarvitse välissä laskuttavaa operaattoria. Mikäli matkapuhelimissa yleistyy WLAN-tekniikka, on se yksi vaihtoehto myös PrintAccess-koodien siirtämiseksi matkapuhelimista digi-TV-päätelaitteille.

## 5.7 Tietoliikenteen sovellustaso

### 5.7.1 Vastakkeet (sockets)

Fyysisen yhteyden ja siirtotien protokollien lisäksi tietoliikennettä käyttävä MHP-sovellus tarvitsee sovellustason protokollan, jotta eri päätelaitteet voivat keskustella keskenään. MHP-standardissa tuetaan paluukanavan tiedonsiirtoprotokollana TCP/IP:tä. Tämän päällä on mahdollista käyttää sovellusprotokollana esimerkiksi HTTP-yhteyksiä (www-sisältö) tai SMTP:tä (sähköposti) /35/, /36/. Nämä yhteydet voidaan toteuttaa Java-ohjelmoinnilla esimerkiksi vastakkeiden (engl. socket) avulla. Vastakkeet abstraktoivat alla olevan tietoliikenteen siten, että sovellus kommunikoi suoraan vastakeluokan ja sille annettavien luku- ja kirjoitusluokkien välityksellä /53/.

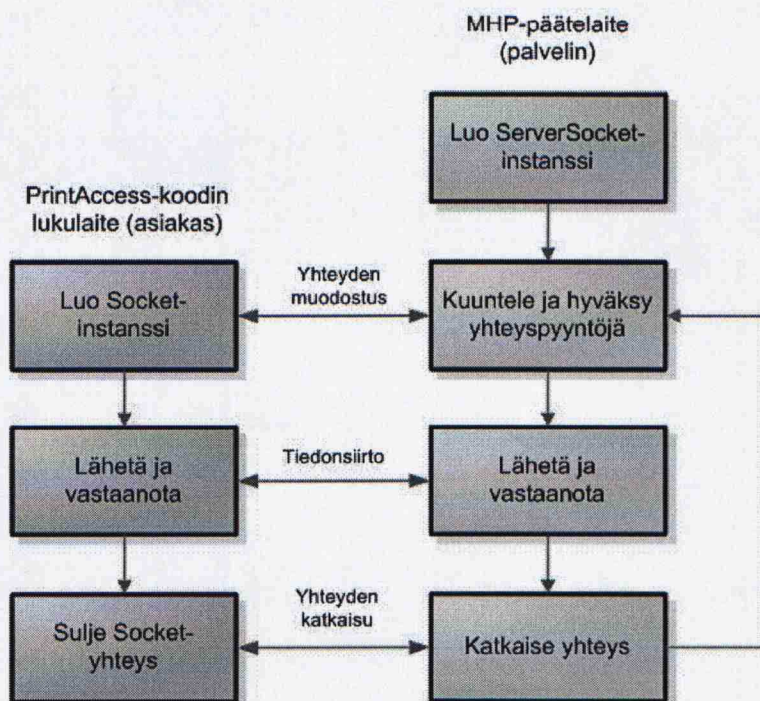
MHP-standardissa oletetaan, että paluukanavan sisältävissä laitteissa kaikki java.net –paketin luokat ovat toteutettuina, joten tällöin markkinoille tulevat, paluukanavan sisältävät ja standardin mukaiset laitteet sisältävät toteutetun java.net.ServerSocket-luokan. Ilman java.net.Socket-luokkaa päätelaitteella ei voida ottaa lainkaan ulkopuolisia paluukanavaa käyttäviä yhteyksiä. /35/, /36/

MHP standardin interaktiivinen ja Internet-profiili sisältävät tuen paluukanavalle, joka voi olla esimerkiksi modeemi, ADSL, ethernet tai vaikkapa WLAN. MHP-standardi ei rajoita

paluukanavassa käytettävää fyysistä verkkorajapintaa kunhan se tukee TCP/IP-liikennettä. Kaikissa yhteystavoissa fyysinen yhteys on luotava ennen kuin vastakeyhteyksiä voidaan käyttää.

MHP-standardi abstraktoi paluukanavan käytön siten, että sovelluskehittäjät voivat käyttää suoraan paluukanavan *RCInterfaceManager* -hallintaluokkaa. Edellämainittuja *java.net.ServerSocket* ja *java.net.Socket* voidaan kuitenkin käyttää myös suoraan. Nämä luokat mahdollistavat TCP/IP-liikenteen, jolloin asiakaslaite voi lähettää informaatiota päätelaitteelle käyttämällä MHP-päätelaitteen IP-osoitetta ja vastakeluokkaan määriteltyä porttia. Portti voi olla mikä tahansa luku 1 ja 65 535:n väliltä. Porttinumerot 1-255 ovat staattisia portteja ja näiden käyttö on kirjattu Internet-suositukseen RFC 1060 /94/. Portit 256-65 535 ovat dynaamisia portteja ja ne ovat sovellusten vapaassa käytössä /1/, /53/.

Otettaessa yhteys MHP-päätelaitteella toimivan *ServerSocket*-instanssin määrittelemään porttiin, päätelaitteella toimiva Java-virtuaalikone käynnistää uuden säikeen, joka suorittaa sille annetun tehtävän. Tehtävä voi olla esimerkiksi lukea lähetetty koodi ja etsiä DVB-lähetyksestä koodia vastaava XML-sisältö. Vastakkeen kautta lähetettävä informaatio voi olla mitä tahansa bittimuotoista informaatiota kuten esimerkiksi ASCII-tekstiä tai kuvia. Kun lähetettävä informaatio on siirretty, yhteys katkaistaan. Kommunikaation vaiheet on esitetty alla olevassa kuvassa (Kuva 15).



**Kuva 15.** Vastakeyhteyden kommunikaation vaiheet /53/



### 5.7.2 Yhdyskäytävät

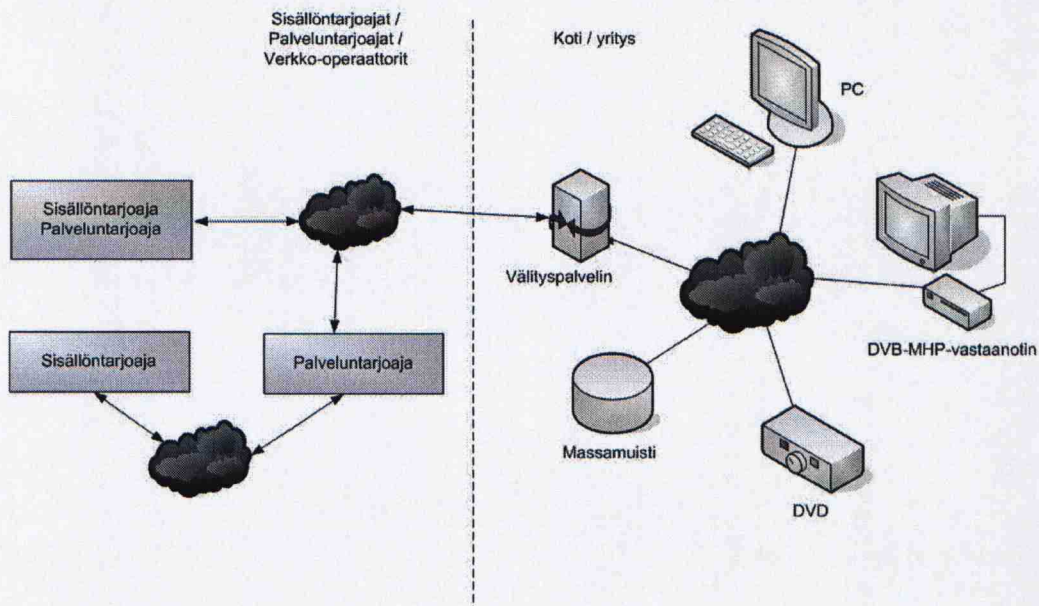
Lukijalaitteen ja MHP-päätelaitteen välinen tietoliikenne voi tapahtua myös epäsuorasti eräänlaisen yhdyskäytävän tai välityspalvelimen välityksellä. Tällainen malli on käytössä kun Internetin sovelluksia käytetään matkapuhelimella WAP-protokollalla (Wireless Application Protocol). WAP-arkkitehtuuri pohjautuu WWW-ratkaisuihin, joita on optimoitu vastaamaan pienten päätelaitteiden vaatimuksia /83/. Matkapuhelinteknologioiden kehityessä ja nopeutuessa myös WAP:lle on tullut lisää käyttömahdollisuuksia ja tekniikka on vähitellen tullut kiinteäksi osaksi useiden matkapuhelinvalmistajien tuotteita.

WAP –teknologiassa Internetin ja matkapuhelinliikenteen välillä toimii yhdyskäytävä eli välityspalvelin, joka vastaa eri protokollien välisestä kommunikaatiosta /83/. Tällainen yhdyskäytävä voidaan toteuttaa myös kodin kulutuselektroniikan yhteyteen, jolloin se välittäisi kommunikaation digi-tv-päätelaitteen ja matkapuhelimen välillä. Tällöin käytettäisiin molemmille laitteille parhaiten sopivia arkkitehtuureja. Matkapuhelin voi kommunikoida välityspalvelimen kanssa langattomasti esimerkiksi WAP-protokollalla käyttäen fyysisenä yhteytenä matkapuhelinverkkoa tai lähiyhteyksissä bluetoothia /48/. Matkapuhelimessa voidaan käyttää WAP:in korvaavana teknologiana myös esimerkiksi J2ME-arkkitehtuuria, jolloin kommunikaatio voidaan suorittaa esimerkiksi RMI-teknologialla /66/.

Välityspalvelin puolestaan kommunikoi päätelaitteen kanssa tilanteesta riippuen langattomasti tai kiinteästi käyttäen esimerkiksi fyysisellä tasolla Ethernet-yhteyttä ja sen päällä jotakin kodin elektroniikan ohjaukseen erityisesti suunniteltua protokollaa. Tällaisia ovat esimerkiksi OSGi, Jini, UpnP, HAVi ja SIP /91/, /92/. MHP:n nykyiset versiot eivät vielä tue näitä teknologioita, mutta tulevat versiot sisältävät todennäköisesti tuen ainakin osalle niistä.

Edellä kuvatun kaltainen teknologia on määritetty Open Service Gateway Initiative (OSGi) –aloitteessa /88/. Kyseessä on yritysyhteenliittymä, joka on vuodesta 1999 asti kehittänyt avointa standardia erilaisten päätelaitteiden kommunikointiin Internetistä lähiverkkoihin. OSGi-mallin mukaisena kehitystrendinä on, että kodin viihdelaitteiden yhteydessä toimii tulevaisuudessa välityspalvelin, joka ohjaa kodin ulkopuolelta tulevia ohjainkomentoja ja laitteiden välistä liikennettä kodin tai yrityksen sisällä /25/, /80/. Tämä on osa kodin lähiverkkojen kehitystä, jossa eri mediaformaatit tallennetaan yhteisiksi resursseiksi ja ne ovat käytettävissä kodin viihdekeskuksilla, joissa on erityisesti viihdekäyttöön suunnitellut käyttöjärjestelmät.

Tallennuskapasiteetti myös kasvaa jatkuvasti ja eri verkoista ladattavaa sisältöä voidaan tallentaa suuria määriä laitteisiin tai niiden oheen asennetuille massamuisteille eli esimerkiksi televisiovastaanottimen digitaaliselle tallentimelle. Seuraavassa kuvassa (Kuva 16) on esitetty OSGi-välityspalvelinmallin mukainen skenaario.



**Kuva 16.** OSGi-välityspalvelinmallin mukainen skenaario /88/, /91/

OSGi:n mukaisessa yhdyskäytävässä käytetään Java Servlet –ohjelmointirajapinnan mukaisia komponentteja, jotka ottavat vastaan yhteyspyyntöjä langattomilta laitteilta /1/, /88/. Servlet:iin voidaan ohjelmoida toiminnallisuus sekä WML- että MIDlet-muotoisille pyynnöille. Tällöin liikenne lähiverkossa kulkee HTTP-protokollaa käyttäen ja yhdyskäytävä muuttaa liikenteen WAP-protokollan mukaiseksi langattomia laitteita varten.

Jini-teknologiassa välityspalvelin voi keskustella MHP-päätelaitteen tai muun kodin elektroniikkalaitteen kanssa Java RMI – teknologiaa (Remote Method Invocation) vastaavalla tavalla kutsuen suoraan päätelaitteella toimivan sovelluksen metodeja /82/. RMI-teknologiaa on selvitetty tarkemmin seuraavassa luvussa. OSGi-yhdyskäytävät voivat periaatteessa sijaita missä tahansa J2ME-yhteensopivassa laitteessa /88/. Tällöin esimerkiksi digitaalisen television päätelaite voi itsessään olla välityspalvelin, jolloin kuluttajan ei tarvitse ostaa erillistä laitetta.

Myös HAVi-standardi määrittelee kommunikaatioprotokollan viihde-elektroniikkalaitteiden väliseen tietoliikenteeseen. HAVi on laajennettavissa eri laitteille FCM –rajapinnan kautta eli jokaisella laitteella tulee olla oma FCM-modulinsa, jonka kautta kommunikaatio tapahtuu. HAVi-mallissa kodin laitteiden väliseen fyysiseen yhteyteen käytetään Firewire-standardia /30/, /51/.



### 5.7.3 RMI

RMI on tekniikka olion metodien etäkutsumiseen. Se on protokolla, joka mahdollistaa olioiden kommunikoinnin eri Java-virtuaalikoneiden välillä sekä resurssien ja prosessointikuorman jakamisen. RMI:n avulla metodit voivat myös siirtää suoraan toisella virtuaalikoneella luotuja olioita toiselle. Virtuaalikoneet voivat sijaita fyysisesti eri tietokoneilla, jolloin kommunikointi tapahtuu tietoliikenneverkon välityksellä /53/. RMI helpottaa sovelluskehittäjän työtä abstraktoimalla sovelluskerrosta alemmat kerrokset siten, että eri olioita voidaan kutsua kuin ne sijaitisivat samalla koneella. RMI huolehtii informaation serialisoinnista ja oikeiden metodien kutsumisesta etäkoneella. Kuljetustason protokollana käytetään yleisimmin TCP/IP:tä, mutta myös suoraan Bluetoothiin päällä olevia RMI-toteutuksia on olemassa. MHP standardin nykyiset versiot tukevat RMI:tä ainoastaan päätelaitteen sisällä Xlet:ien väliseen kommunikaatioon /79/. Ulkopuolista yhteyttä varten tarvitaan erillinen komponentti, joka kuuntelee RMI-kutsuja.

Käytettäessä RMI-tekniikkaa tulee MHP-päätelaitteella olla toiminnassa sovellus, joka tarjoaa ulkopuolisille yhteyksille rekisterin päätelaitteella olevista RMI-metodeista. Tämä rekisteri on saatavissa RMI-palvelimena toimivalta koneelta RMI-kyselyllä. Olennaisesti RMI koostuu kolmesta kerroksesta: Alimpana on kuljetuskerros, jonka päällä on rekisterikyselyihin vastaava etäreferenssikerros. Tämän päällä on edustajakerros, joka koostuu Proxy-suunnittelumallin mukaisista edustajaluokista /45/. Kerrosmalli mahdollistaa eri kerroksien muuttamisen vaikuttamatta muuhun systeemiin kuten esimerkiksi vaihdettaessa TCP/IP-protokollasta suoraan Bluetoothiin /53/.

RMI:n toimintamallissa erotetaan itse toteutus ja toteutuksen kuvaus toisistaan. Toteutuksen kuvaus asetetaan edustajakerrokseen saataville ja etäkone kysyy kuvauksia toteutuksista referenssikerroksen kautta. Kun asiakkaalla on käytössään palvelinohjelman toteutuksen kuvaus, se voi kutsua metodeita kuten ne sijaitisivat samalla koneella. RMI huolehtii siitä että kutsuminen siirretään tietoliikenneverkon ylitse ja varsinainen toteutus tapahtuu palvelinkoneella. /1/, /53/

## 6 SOVELLUSKEHYKSET

### 6.1 Yleistä

Oliopohjaisen ohjelmistosuunnittelun kehityksen alusta lähtien sen keskeisiä tavoitteita on ollut helpottaa ohjelmistosuunnittelua ja nopeuttaa ohjelmistotuotantoa mahdollistamalla komponenttien uudelleenkäyttö. Tämänkaltaisen suunnittelu on mahdollistanut erilaisiin liiketoimintaprosesseihin erikoistuneiden sovelluskehysten laatimisen. Sovelluskehyksellä tarkoitetaan yleisesti uudelleenkäytettäväksi suunniteltua systeemiä, joka koostuu abstrakteista luokista ja säännöistä joiden mukaan näistä luodut ilmentymät toimivat keskenään /39/. Ohjelmoija käyttää sovelluskehystä laatimalla oman sovelluksensa käyttäen näitä luokkia ja niiden tarjoamia valmiita palveluita.

Sovelluskehysten laatiminen on kalliimpaa ja vaativampaa kuin perinteinen sovelluskehitys, eikä se siten ole itsessään ratkaisu ohjelmistokehityksen helpottamiseksi. Parhaimmillaan oikein toteutettu sovelluskehys voi kuitenkin vähentää merkittävästi tietyn alueen sovellusten valmistuskustannuksia ja vähentää virheitä sovelluskehysten avulla laadituissa ohjelmistoissa /39/. Kehittyneet sovelluskehykset voivat vähentää oman alueensa sovellusten tekemiseen käytettävää lähdekoodia jopa 90% verrattuna sovellusten tekemiseen pelkkiä standardiluokkakirjastoja käyttäen /38/, /39/, /28/. On kuitenkin huomattava, että kehittyneen sovelluskehysten laatimiseen vaadittavat panostukset ovat usein moninkertaisia yksittäisen sovelluksen tekemiseen verrattuna. Tästä syystä sovelluskehysten tekeminen edellyttää että

- samalla sovellusalueella tullaan toteuttamaan useita samankaltaisia sovelluksia ja
- sovellusalueen tekniikasta ja sovelluskehityksestä on riittävästi tuntemusta.

Oliopohjaisten sovelluskehysten suunnitteluperiaatteina pidetään modulaarisuutta, uudelleenkäytettävyyttä, laajennettavuutta ja kontrollin siirtämistä /39/. Nämä termit on selitetty tarkemmin alla.

**Modulaarisuus:** Sovelluskehykset kätkevät sovellusalueen usein muuttuvia yksityiskohtia pysyvien rajapintojen taakse. Tällöin voidaan tehdä muutoksia yksityiskohtiin ilman että muita sovelluksen osia tarvitsee muuttaa samalla. Tämän tyyppinen suunnittelu vähentää myöhemmin tarvittavaa panostusta sovelluksen ylläpitoon.

**Uudelleenkäytettävyys:** Sovelluskehysten tarjoamat rajapinnat lisäävät uudelleenkäytettävyyttä määrittämällä geneerisiä komponentteja, joita voidaan hyödyntää luotaessa uusia sovelluksia. Geneeristen komponenttien avulla sovellusalueita koskeva erityistieto ja kokeilempien ohjelmistosuunnittelijoiden panostus voidaan välittää uusien sovellusten tekijöille ilman että ohjelmiston samoja elementtejä tarvitsee luoda uudelleen. Tämänkaltaisen uudelleenkäytettävyys mahdollistaa huomattavan tuottavuuden parannuksen ohjelmistokehityksessä ja samalla parantaa uusien ohjelmien laatua ja suorituskykyä.



**Laajennettavuus:** Sovelluskehityksen tulee olla laajennettavissa jotta mahdolliset muutokset liiketoimintalogiikassa saadaan päivitettyä sovelluskehitykseen. Laajennettavuus mahdollistetaan tarjoamalla erityisiä alkeismetodeja (hook method – metodi joka ei käytä muita metodeja), jotka mahdollistavat sovelluskehityksen rajapintojen laajentamisen sovelluksesta käsin. Rajapintojen laajentaminen tapahtuu nk. kuumien ja jäädytettyjen pisteiden avulla (hot spots ja frozen spots). Kuumat pisteet ovat abstrakteja luokkia tai rajapintoja, jotka sovelluksen täytyy toteuttaa käyttäessään sovelluskehitystä. Tiedot osat sovelluskehityksestä ovat pysyviä toteutuksia ja näitä kutsutaan jäädytetyiksi pisteiksi. Jäädytetyt osat muodostavat sovelluskehityksen ytimen, joka kutsuu sovelluskehitystä käyttävän sovelluksen rajapintoja.

**Kontrollin siirtäminen:** Kontrollin siirtämisellä tarkoitetaan, että arkkitehtuuri mahdollistaa tapahtumankäsittelijäolioiden kutsumisen sovelluskehityksen kautta. Kun jokin tietty tapahtuma tapahtuu, sovelluskehitys kutsuu edellä mainittuja alkeismetodeja, jotka on toteutettu sovelluksessa sovellusohjelmoijan määrittelemällä tavalla. Tällä tavoin mahdollistetaan, että sovelluskehitys päättää mitä sovelluskohtaisia metodeja kutsutaan missäkin tilanteessa.

#### 6.1.1 Lasilaatikko- ja mustalaatikkokomponentit

Sovelluskehitykset eroavat pelkistä luokkakirjastoista siten, että ne määrittävät toiminnallisuuden lisäksi myös komponenttien välisen vuorovaikutuksen /39/, /74/. Luokkia, joiden metodeita ei ole täydellisesti toteutettu, kutsutaan lasilaatikkokomponenteiksi. Tällä tarkoitetaan luokkia, jotka ohjelmoijan tulee syrjäyttää (override) rakentaessaan sovellusta. Syrjäyttämistarpeen vuoksi sovellusohjelmoijan on välttämätöntä tuntea riittävästi sovelluskehityksen toiminnallisuutta ja rakennetta, jotta tarvittavat metodit tulevat toteutettua. Tämän vuoksi sovelluskehityksen ja sen kuumien pisteiden dokumentointiin tulee kiinnittää kehitysvaiheessa erityistä huomiota.

Mustalaatikkokomponenteilla tarkoitetaan luokkia, joissa metodeille on valmiit toteutukset. Näitä komponentteja ohjelmoija voi käyttää suoraan omassa toteutuksessaan ja muutokset niihin tehdään kompositioiden avulla ja muuttamalla komponentin parametreja /39/.

Sovelluskehityksessä on yleensä sekä lasilaatikko-, että mustalaatikkokomponentteja. Mitä enemmän kehitystä käytetään, sitä enemmän tunnistetaan tarpeita siirtää tiettyjen lasilaatikkokomponenttien toteutus mustalaatikkokomponentteihin. Tästä syystä pitkälle kehitetyissä sovelluskehityksissä painopiste on siirtynyt enemmän mustalaatikkokomponenttien puolelle.

#### 6.1.2 Suunnittelumallit

Olio-ohjelmoinnin arkkitehtuuri mahdollistaa tiettyjen ongelmakenttien ratkaisemisen aina samankaltaisella tavalla. Tätä seikkaa tukemaan on syntynyt suuri joukko erilaisia suunnittelumalleja, jotka edustavat kukin tietynlaisen ohjelmointitilanteen hyväksi havaittua toteu-

tusta. Suunnittelumallit (engl. design patterns) ovat testattuja ratkaisuja ja standardimuotoisuutensa ansiosta helpottavat kommunikointia ohjelmoijien välillä. Suunnittelumallit eivät ole riippuvaisia ohjelmointikielestä, mutta liittyvät yleensä vahvasti olio-ohjelmointiin./45/, /47/

Sovelluskehysten laatiminen liittyy läheisesti suunnittelumalleihin, sillä tietyssä mielessä sovelluskehykset voidaan nähdä tiettyyn ongelmakenttään sidottujen suunnittelumallien toteutuksena /39/. Sovelluskehysten suunnittelussa tulisikin käyttää suunnittelumalleja aina kun se on mahdollista sovelluskehysten käytön ja jatkokehityksen helpottamiseksi.

## 6.2 Sovelluskehysten laatiminen

Sovelluskehysten laatiminen eroaa perinteisestä sovelluskehityksestä eniten siinä että pitkälle kehittyneen sovelluskehysten tulee kattaa kaikki sovellusalueensa erityispiirteet siinä missä sovelluksen tulee kattaa vain sille asetetut vaatimukset /28/, /39/. Erityispiirteiden kartoittaminen on mahdotonta ilman ongelmakentän syvällistä tuntemusta. Tästä johtuu, että sovelluskehysten laatiminen on iteratiivinen prosessi, jossa sovelluskehysten lisätään ominaisuuksia sitä mukaa kun ongelmakentän tuntemus paranee /27/. Sovelluskehysten laatimisessa edetään yleisesti neljässä vaiheessa /74/. Nämä vaiheet ovat

1. Ongelmakentän analyysi
2. Sovelluskehysten suunnittelu
3. Sovelluskehysten toteutus
4. Sovelluskehysten "alustaminen" eli sovellusten rakentaminen sovelluskehysten avulla.

**Ongelmakentän analyysi** pyrkii selvittämään sovellusalueen nykyiset ja mahdolliset tulevaisuuden vaatimukset eli vaihetta voi myös kutsua vaatimusmäärittelyksi. Analyysivaiheessa tulee selvittää sovelluskehysten toteutettavat kuumat ja jäädytetyt pisteet.

**Sovelluskehysten suunnitteluvaiheessa** suunnitellaan ja mallinnetaan sovelluskehysten arkkitehtuuri. Abstraktiot, kuten abstraktit luokat ja tarvittavat rajapinnat kuvataan arkkitehtuurin kuvauskielillä kuten UML-kaavioilla. Tässä vaiheessa päätetään sovelluskehysten käytettävät suunnittelumallit ja näiden rakenne mallinnetaan.

**Sovelluskehysten toteutusvaiheessa** laaditaan kehysten ohjelmakoodi edellä tehdyn suunnittelun mukaisesti.

**Sovelluskehysten alustuksella** tarkoitetaan sovelluksen tekemistä käyttäen sovelluskehystä. Tässä vaiheessa sovelluskehysten rajapinnat ja abstraktiot toteutetaan ja laaditaan ajettava sovellus. Kaikille sovelluskehysten sovelluksille on yhteistä kehysten ydin eli jäädytetyt pisteet. Alustuksessa sovelluskehysten koodista löydetty virheet korjataan ja kehyksestä laaditaan uusia versioita tarpeen mukaan.



Alustusvaiheessa testataan sovelluskehystä käyttöolosuhteita vastaavasti. Sovelluskehysten laatiminen ja sen testaaminen on kokonaisuudessaan jatkuva ja iteratiivinen prosessi, sillä vain harvoissa tapauksissa sovellusalueen vaatimukset tunnetaan ennaltakäsin niin tarkasti, että sovelluskehyksestä voitaisiin tehdä vain yksi pitkäaikainen versio /39/. Useimmiten sovelluskehys elää siinä missä sen avulla rakennettavat sovelluksetkin ja siihen kehykseen lisätään ominaisuuksia sitä mukaa kun tarvetta ilmenee. Tämä tulee ottaa huomioon myös sovelluskehysten ensimmäistä versiota laadittaessa.

Sovelluskehysten tekeminen eroaa muusta oliosuuntautuneisesta kehityksestä siten, että siinä missä yleisesti oliopohjaisessa ohjelmistokehityksessä tutkitaan jotakin tiettyä ongelmaa ja laaditaan siihen ratkaisu, täytyy sovelluskehysten tarjota konseptitasolla vastaukset koko ongelmakentän vaatimuksiin. MHP-ympäristön erikoispiirteenä on rajallinen suorituskäytöskäytäntä /111/. Suunnitelluista PrintAccess-konsepteista taas on puolestaan etsittävä yhteiset nimittäjät ja koottava nämä sovelluskehysten palveluiksi.

Sovelluskehysten voidaan katsoa olevan hyvin suunniteltu, mikäli se tarjoaa ongelmanratkaisulle oleelliset kuumat pisteet /39/. Tällöin se määrittelee sovelluskehystä käyttävän sovelluksenkin arkkitehtuurin eli kompositiot ja komponenttien vuorovaikutuksen. Sovelluskehysten avulla rakennetut sovellukset eivät ainoastaan käytä kehyksen lähdekoodia vaan myös sen arkkitehtuuria omassa toiminnassaan /41/.

MHP-ympäristöön rakennettavan PrintAccess-sovelluskehysten tarkoituksena on helpottaa sovelluskehitystä ja tarjota sovellusohjelmoijalle valmiit välineet ko. ympäristön nopeaan käyttöönottoon. Tällöin myös sovelluksen käynnistämiseen ja *Xlet*-rajapinnan käyttöön liittyvät tekniset seikat on syytä tuoda sovelluskehysten puolelle ja jättää sovellusohjelmoijan huoleksi ainoastaan oman sovelluksensa ulkoasu ja toimintalogiikka. Sovelluskehysten arkkitehtuurikysymyksiin paneudutaan tarkemmin seuraavassa luvussa.

## 7 PRINTACCESS SOVELLUSKEHYKSEN LAATIMINEN MHP-YMPÄRISTÖÖN

### 7.1 Yleistä

Diplomityössä DVB-MHP-standardin soveltuvuutta PrintAccess-tarkoituksiin tutkitaan käytännössä kehittämällä MHP-teknologiaa käyttävän PrintAccess-sovelluskehityksen ensimmäinen versio ja testaamalla sen toimivuutta valitun konseptin mukaisella sovelluksella. Sovelluskehityksen perusajatuksena on tuoda PrintAccess –konseptien ja sovellusten kehittäminen korkeammalle tasolle, jolloin jatkossa tämän alueen sovellusten testaaminen MHP-ympäristössä on nopeampaa ja vaatii vähemmän resursseja. Sovelluskehitykselle laaditaan vaatimusmäärittely, jonka pohjalta tehdään arkkitehtuurisuunnittelu. Arkkitehtuurin ilmaisemisessa käytetään UML-standardin mukaisia käytäntöjä /21/. Luokkanimet ovat englanniksi yleisten ohjelmointikäytäntöjen mukaisesti /43/. Laaditun suunnittelun pohjalta sovelluskehitys toteutetaan laatimalla suunnitelmaa vastaavat luokat ja rajapinnat.

### 7.2 PrintAccess-sovelluskehityksen vaatimusmäärittely

Edellä olevissa luvuissa on käyty läpi MHP-teknologiaa ja PrintAccess-sovellusten erityispiirteitä. Jotta sovelluksia voitaisiin toteuttaa MHP-ympäristössä, on ympäristöstä löydettävä ratkaisut niiden teknisiin vaatimuksiin. Nämä vaatimukset voidaan jakaa primäärisiin ja sekundäärisiin. Primäärisillä vaatimuksilla tarkoitetaan kaikille PrintAccess-sovelluksille välttämättömiä toiminnallisuuksia kuten koodisyytöiden ja sovelluksen sisällön käsittelyä sekä käyttöliittymän toimintoja. Sekundäärisillä vaatimuksilla tarkoitetaan PrintAccess-konseptien toteuttamisen kannalta hyödyllisiä toiminnallisuuksia kuten yleiskäyttöisiä käyttöliittymäkomponentteja. Seuraavaan taulukkoon on koottu PrintAccess-sovelluskehityksen primääriset ja sekundääriset vaatimukset. Primäärisiin vaatimuksiin on löydyttävä ratkaisut, jotta PrintAccess-sovellusten toteuttaminen MHP-ympäristössä on millään tasolla mahdollista. Tällainen vaatimusmäärittely, jossa pyritään hakemaan sovellusalueen yleiset ongelmat vastaa sovelluskehysten suunnitteluprosessissa käyttötapausten etsimistä, joka puolestaan on yleinen menetelmä oliosuuntautuneessa ohjelmistokehityksessä ohjelmiston vaatimusten määrittelemiseksi /39/, /69/.

Seuraavassa taulukossa (Taulukko 6) esitetään kootusti PrintAccess-sovellusten vaatimukset ja sovelluskehitykseen ohjelmoitava, ko. vaatimuksen täyttävä ominaisuus. Diplomityön puitteissa laadittava PrintAccess-sovelluskehityksen ensimmäinen iteraatio koostuu näistä ominaisuuksista.



**Taulukko 6. PrintAccess-sovellusten vaatimukset ja sovelluskehiksen ominaisuudet**

|    | <b>PrintAccess-vaatimus</b>   | <b>Toteutettava ominaisuus sovelluskehikseen</b>   | <b>Prioriteetti (Primäärinen / sekundäärinen)</b> |
|----|---|--|---|
| 1. | PrintAccess-sovellukset toimivat saman sovelluskehiksen alla  | Sovellusten käynnistäminen yhtenäisestä käyttöliittymästä  | Primäärinen                                       |
| 2. | Sovellusten ohjelmointi helpottuu ja ne toimivat yhdenmukaisesti  | Sovelluskehys huolehtii sovelluksen tilan hallinnasta  | Primäärinen                                       |
| 3. | Informaatiota voi lähettää lukijalaitteelta MHP-päätelaitteelle kaukosäätimen yhteyskäytännöllä (infrapuna)   | PrintAccess-koodin lukulaitteelta tulevan koodisyötteen tulkitseminen käyttäen ohjainlaitteille määriteltä <i>UserEventListener</i> -yläluokkaa.   | Primäärinen                                       |
| 4. | Informaatiota voi lähettää lukijalaitteelta MHP-päätelaitteelle lähiverkon yhteyskäytännöllä (ethernet, bluetooth, WLAN)  | PrintAccess-koodin lukulaitteelta tulevan koodia vastaavan syötteen lukeminen käyttäen paluukanavan olemassaolevaa <i>Socket</i> -luokkaa ja TCP/IP protokollaa.   | Primäärinen                                       |
| 5. | Sovellusten sisältö on helposti ylläpidettävissä ja sisältötiedostot ovat erikseen lähetettävissä DVB-yhteydellä.   | XML-muotoisen sisällön lataus DVB/MHP-lähetyksestä ja sisällön käsittely.  | Primäärinen                                       |
| 6. | Sovellusten käyttöliittymien laatiminen onnistuu yhtenäisellä tavalla ja sovelluskehys tarjoaa yleisimmät käyttöliittymäkomponentit valmiiksi ohjelmoijan käyttöön. | Käyttöliittymän komponenttien hallinta toteutetaan tunnetulla ja laajennettavalla arkkitehtuurilla. Sovelluskehys sisältää yleiset käyttöliittymäkomponentit kuten tekstikentät, jotka ohjelmoija voi ottaa käyttöön vain parametrisoimalla ne haluamallaan tavalla. | Primäärinen                                       |
| 7. | Sovellusten sisällön käsittely on yhtenäistä eri sovellusten välillä.   | Sovelluskehys tarjoaa XML-tiedon lukemiseen ja kirjoittamiseen tarvittavat rajapinnat  | Primäärinen                                       |

|     |   |   |               |
|-----|---|---|---------------|
| 8.  | Sovellusten muuttuvien tietojen (esim. IP-osoitteet yms.) ylläpito on keskitetty yhteen paikkaan  | Konfiguraation hallintaan on yksi tiedosto, josta muutetaan kaikkien sovellusten muuttujia.                     | Sekundäärinen |
| 9.  | Sovellusten virheidenkorjausta on helpotettu MHP-ympäristöä ajatellen.  | Sovelluskehys tarjoaa poikkeusten hallintaan valmiit luokat, jotka tulostavat informatiivisia virheilmoituksia. | Sekundäärinen |
| 10. | Koodi-informaatio on mahdollista lähettää suoraan kuvamuodossa päätelaitteelle, jotta mahdollisten dekodausalgoritmien toteutus MHP-ympäristössä on mahdollista | Sovelluskehys tarjoaa lukijalaitteelta tulevan kuvamuotoisen koodin lukemiseen tarvittavat yläluokat.           | Sekundäärinen |

## 7.3 PrintAccess-sovelluskehysten arkkitehtuuri

### 7.3.1 Yleinen arkkitehtuuri

Sovelluskehysten suunnittelussa pyritään etsimään sopivat arkkitehtuuriset ratkaisut edellisessä luvussa mainittujen vaatimusten toteuttamiseksi. Ratkaisuissa pyritään soveltamaan yleisiä käytäntöjä, jotta sovelluskehysten ylläpito jatkossa olisi mahdollista. Tästä syystä sovelluskehysten yleinen arkkitehtuuri toteutetaan noudattaen MHP-ympäristöön soveltuvin osin kolmitasoista MVC (Model, View, Controller – sisältö, näkymä ja ohjain) – suunnittelumallia. Kyseinen malli on yleisesti käytetty ratkaisu graafisen käyttöliittymän sisältävissä sovelluksissa, jossa sovellus jaetaan osiin siten että käyttäjän syötteiden vastaanottaminen, käsittely ja ohjelman näkyvä toiminta ovat kaikki erillisiä kokonaisuuksia /27/, /73/.

MVC:ssä eristetään tietosisältö (model) sen esittämisestä (view) ja siihen kohdistuvan käsittelyn hallinnasta (controller). Ohjain tulkitsee käyttäjän syötteet komennoiksi, jotka tulkitaan mallissa olevan logiikan mukaisesti. Tämän jälkeen tulokset esitetään näkymässä.

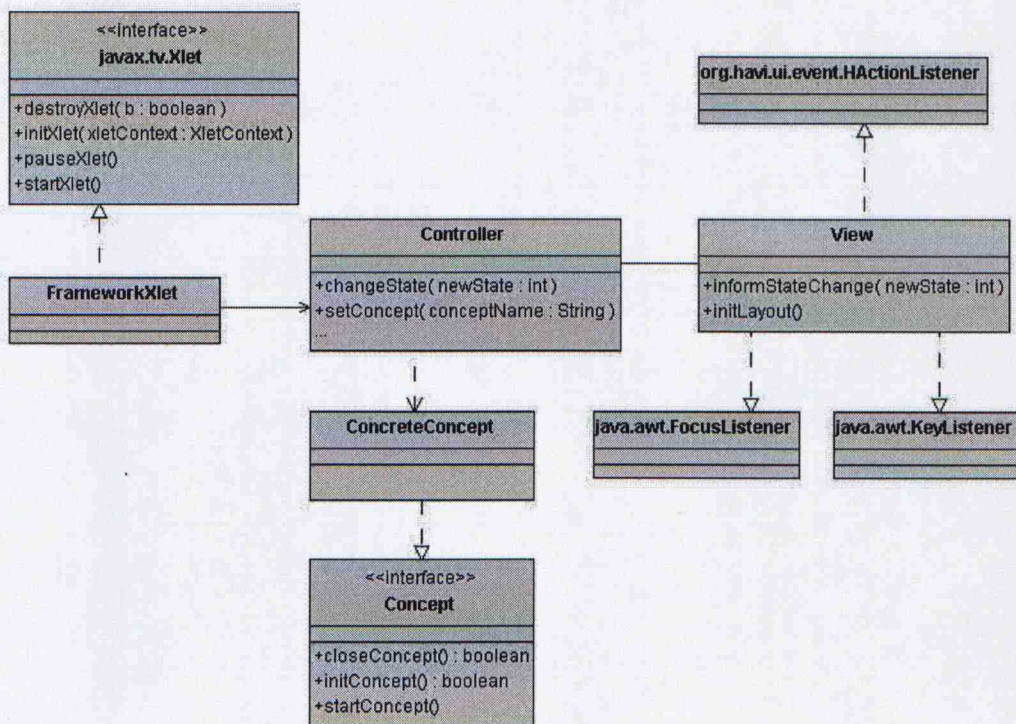
Ohjain on tietoinen sovelluskehysten tilasta ja informoi tilan muutoksista näkymää, joka muuttaa käyttöliittymää uutta tilaa vastaavasti /27/, /32/. Sisältöoliot pitävät puolestaan sisällään kaikki esitettävät tiedot ja tietojen muuttamiseen tarvittavat metodit.

Sisältöoliot voidaan pitää täysin tietämättöminä käyttöliittymästä ja tietojen esittämismuodosta. Tämä on ylläpidettävyyden kannalta kestävä ratkaisu, sillä käyttöliittymää ei tarvitse muuttaa vaikka sisältö muuttuisikin. PrintAccess-sovelluskehysten tapauksessa tämä tar-



koittaa, että sovellusohjelmoijan on lisättävä ainoastaan uusi tietosisältö luodakseen uuden sovelluksen.

MHP-ympäristö eroaa PC-pohjaisesta käyttöliittymäympäristöstä siten, että päätelaite suorittaa *Xlet* -rajapinnan toteuttavan luokan sovellusta käynnistettäessä. Tämän vuoksi sovelluskehiksen *FrameworkXlet*-luokka on vastuussa sovelluskehiksen ohjaimen alustamisesta. Sovelluskehiksessä ohjain liitetään näkymään, jolloin näkymä toteuttaa tapahtuman-kuuntelijat *KeyListener*, *HActionListener* ja *FocusListener*. Rakenne on esitetty UML-luokkakaaviona seuraavassa kuvassa (Kuva 17). Sovelluskohtainen tietosisältö on aina sovellusohjelmoijan vastuulla. Ohjain käsittelee sovelluksen tietosisältöä *Concept*-rajapinnan kautta siten, että sovelluskehys osaa käynnistää aina oikean sovelluksen (yleisnimitys sovellukselle: *ConcreteConcept*). Konkreettisen sovelluksen laatiminen on aina sovelluskehittäjän vastuulla. Sovellusten käynnistäminen on selitetty tarkemmin seuraavassa luvussa.



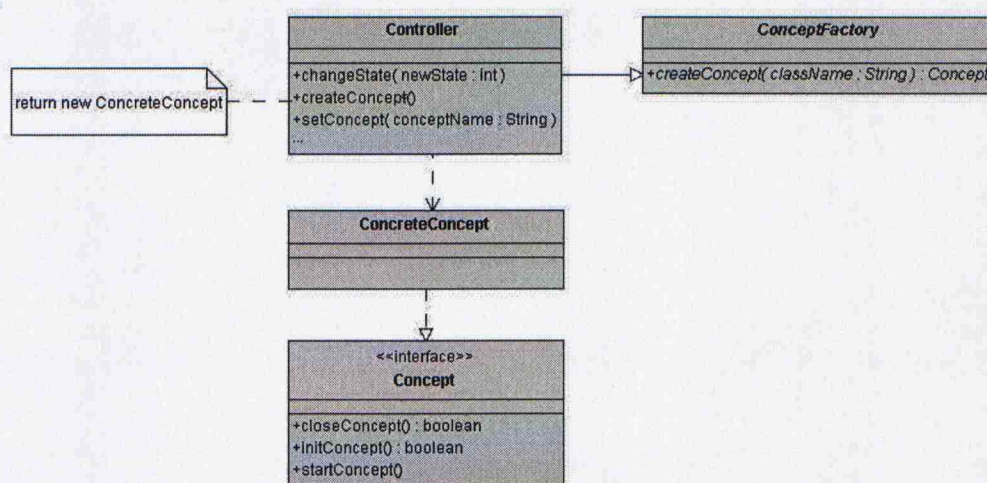
**Kuva 17.** Sovelluskehiksen rakenne MVC-arkkitehtuurin mukaisesti

### 7.3.2 PrintAccess-sovellusten käynnistäminen sovelluskehiksellä

Sovelluskehys tarjoaa palvelun sovellusten käynnistämiseen MHP-päätelaitteessa. Tällöin sovelluskehys tulee ensin käynnistää, jotta sovelluksia voitaisiin ajaa. Tässä MHP-sovel-

luskehys eroaa perinteisestä sovelluskehysrakenteesta, jossa sovelluskehys ei itsessään ole suoritettavissa /39/. Sovelluskehysen käynnistettävyyden vuoksi sovelluskehys tarjoaa *Xlet*-rajapinnan toteutuksen, jonka avulla käynnistys päätelaitteella tapahtuu. *FrameworkXlet*-olio luodaan päätelaitteen hallintaohjelmistossa. Olio luo käynnistämisensä yhteydessä instanssin sovelluskehysen ohjaimesta (*Controller*), joka puolestaan on vastuussa sovellusten esittämisestä käyttäjälle.

Tehdasmetsodi (Factory Method) on yleisesti sovelluskehyksissä käytetty suunnittelumalli, jonka avulla luokka voi siirtää olioiden luonnin aliluokille /45/. Tällöin aliluokkien tehtävänä on päättää mistä luokasta ilmentymä luodaan. Tällainen toiminnallisuus on käyttökel-poinen sovelluskehyksissä, koska se mahdollistaa aiemmin mainitun sovelluskehysten vastuun siirtämisen. Vaikka sovelluskehys on vastuussa olioiden luomisesta, ei sitä laadittaessa tarvitse tietää mistä konkreettisesta luokasta olioita luodaan. Sovelluskehys käyttää ainoastaan konkreettisille luokille yhteistä rajapintaa. Tämän mallin mukainen luokkakaa-vio on esitetty seuraavassa kuvassa (Kuva 18).



**Kuva 18.** Sovellusten käynnistäminen oliotehdas-suunnittelumallin mukaan /45/

PrintAccess-sovelluskehyksessä ohjainluokka toteuttaa oliotehdas-rajapinnan, jolloin se pystyy pyynnöstä luomaan uuden konkreettisen konseptin. Tällainen pyyntö annetaan käyttäjän toimesta kun käyttäjä valitsee haluamansa konseptin esimerkiksi sovelluskehysen alkuvalikosta.

### 7.3.3 Sovelluskehysen tilan hallinta

Sovelluskehysen ohjaimen tulee tietää kullakin ajan hetkellä suoritettavan sovelluksen tila, jotta se voi reagoida eri tapahtumiin oikein. Eri tiloja ovat sovelluskehysen alustaminen, sovellusten käynnistäminen ja pysäyttäminen, valikkotila, eri käyttöliittymäkompo-



nenttien esittäminen sekä PrintAccess-koodien lukeminen ja prosessoiminen. Siirtyminen sovelluskehyksestä suoritettaviin konsepteihin tapahtuu alkuvalikon kautta. Käyttäjän syöteistä johtuvat tilan muutokset käsitellään *View*-luokan *informStateChange*-metodissa. Tällä metodilla *Controller*-oliolle kerrotaan uuden tilan tunnus. *Controller*-olion *changeState* muuttaa sovelluskehysten tilan uutta tunnusta vastaavasti.

#### 7.3.4 Konfiguraation hallinta ja poikkeusten käsittely

Konfiguraation hallinta on oleellinen osa sovelluskehysten toimintaa. Koska MHP-ympäristössä *Xlet*-luokka on vastuussa sovellusten käynnistämisestä, on sovelluskehyksellä oltava tiedot kaikista sen kautta käynnistettävistä sovelluksista. Ohjelmoijien tulee voida helposti muuttaa sovelluksen parametreja ja sovellukselle on tarjottava oikeat tiedot sen ympäristöstä.

Sovelluskehysten konfiguraatietietoja ovat esimerkiksi sovelluskehysten tilojen tunnukset, sisältötiedostojen osoitteet ja sovelluskehyksellä suoritettavien sovellusten nimet. Ylläpidettävyyden vuoksi näitä konfiguraatietietoja säilytetään keskitetysti. Tämä onnistuu kuvaamalla sovelluksen parametrit avain-arvo –pareina XML-tiedostossa. Sovelluksen suorituskyvyn kannalta on oleellista, että konfiguraatietiedot siirtyvät sovellusta käynnistettäessä välimuistiin, josta ne on luettavissa nopeasti. Tämä toteutetaan lukemalla tarvittavat avain-arvo –parit sovellusta käynnistettäessä konfiguraatietiedostosta sopivaan *HashTable* –olioon. Lukemisesta huolehtii sovelluskehysten *Configuration*-luokka. Konfiguraatiomallin mukaista XML –parserointia voidaan käyttää myös yleisesti sovelluskehysten tarvitseman XML-muotoisen tiedon parsimiseen kuten esimerkiksi silloin kun sovelluksen sisältö toimitetaan erillisinä XML-tiedostoina.

Sovelluskehys tarjoaa poikkeusten käsittelyyn oman *FrameworkException* –luokan. Tämä luokka periytyy *java.lang.Exception*-luokasta ja se heitetään sovelluksessa eri virhetilanteiden tapahtuessa. Koska virheiden paikantaminen koodista on MHP-ympäristössä melko hankalaa, annetaan *FrameworkException*-oliolle parametreina varsinainen kiinniotettu poikkeus ja lyhyt kuvaus poikkeustilanteesta, jonka olio voi tulostaa konsoliin virhekohdan löytämiseksi. Tämä informaatio on lopullisen sovelluksen kannalta tarpeellista, joten luokassa tulee olla keino, jolla informaation tulostuksen voi kytkeä helposti pois päältä. Tarkoitusta varten sovelluskehys laaditaan *Debug* –luokka, joka sisältää metodit erilaisten informaatio- ja poikkeusviestien tulostamiseen. Molemmat ominaisuudet voidaan kytkeä pois päältä.

#### 7.3.5 XML-informaatio ja sen tulkitseminen

Luetut koodit täytyy pystyä yhdistämään palvelun sisältöön esimerkiksi siten, että luettu koodi vastaa XML-muotoisen sisältötiedoston tiettyä tagia, joka sitten parsitaan käyttäjälle nähtäväksi. Mikäli päätelaitteeseen luettava informaatio on rajattu palveluntarjoajan osana DVB-lähetystä lähettämään informaatioon, on XML-muotoisten sisältötiedostojen määrä varsin rajattu oliokarusellin rajoitusten vuoksi. Tällöin kovin laajamuotoisia palveluja ei



ole mahdollista toteuttaa. Esimerkiksi Suomen digitelevisiossa käytössä oleva superteksti-televisiosovellus on toteutettu XML-sisällönkuvaustiedostojen avulla /95/. Jatkossa MHP-päätelaitteiden paluukanavaratkaisujen kehittyessä sovellustiedostoja voidaan yhä enemmän ladata paluukanavayhteydellä. Tällöin on mahdollista ladata enemmän sisältötiedostoja ja sitä kautta muodostaa laajempia sovelluksia.

PrintAccess-sovelluskehys sisältää metodit XML-informaation yksinkertaiseen tulkitsemiseen (parsiminen). Nämä metodit on toteutetaan *XMLHandler*-luokassa. Koska MHP-päätelaitteissa on varsin rajallinen määrä suorituskapasiteettia, käyttää *XMLHandler*-luokka ohjelmallisesti käsiteltävien XML-elementtien muodostamiseen valmista avoimen lähdekoodin NanoXML-parseria, joka on optimoitu pienen muistimäärän omaaville laitteille /33/.

### 7.3.6 Käyttöliittymäkomponenttien hallinta

Käyttöliittymän hallinnasta vastaa sovelluskehiksen *View*-luokka (näkymä). Ohjain informoi näkymää sovelluskehiksen tilan muutoksista *informStateChange* –metodilla. Tälle metodille annetaan parametrina uuden tilan numero. Sovelluskehiksen näkymä-luokka perii *org.havi.ui.HContainer*-luokasta, joten se voidaan lisätä Xletin *HScene*-olioon. *View*-olioon voidaan lisätä käyttöliittymäkomponentteja *addComponent* –metodilla, jolle annetaan parametreina lisättävä komponentti, komponentin koko ja komponentin sijainti *HContainer*:ssa XY-koordinaateilla. Tätä metodia käyttäen sovellusohjelmoijat voivat koostaa sovelluksensa käyttöliittymän. *View*-luokassa ei rajoiteta lisättävää komponenttia muutoin kuin että sen on perittävä *Component*-luokasta. Esimerkkejä lisättävistä komponenteista ovat painikkeet, valintaruudut ja tekstikentät.

Sovelluskehys tarjoaa ohjelmoijalle joitakin valmiita käyttöliittymäkomponentteja. Nämä löytyvät sovelluskehiksen *fi.hut.media.printaccess.framework.ui*-paketista. Esimerkiksi kaukosäätimellä tapahtuvaan koodien lähetykseen on sovelluskehyksessä oma lukijakomponenttinsa nimeltä *CodeReader*.

## 7.4 Tietoliikenne

### 7.4.1 PrintAccess-koodien lukeminen

PrintAccess-sovelluksissa painotuotteesta luetut koodit tulkitaan joko lukulaitteella tai jollakin ulkopuolisella palvelimella kuten luvussa 5.2 mainittiin. MHP-päätelaitteelle koodissa oleva informaatio tulee tulkitussa muodossa merkkijonona joko kaukosäätimen komentoina tai paluukanavan yhteyttä käyttäen. Paluukanavan yhteyksiin tarvitaan käytännöt sekä päätelaitteelta ulkopuoliseen verkkoon, että ulkopuolisilta laitteilta kuten PrintAccess-koodien lukijalaitteilta päätelaitteelle otettavien yhteyksien hallintaan. Viimeisin / viimeisimmät luetut merkkijonot annetaan *CodeStorage*-oliolle, joka huolehtii niiden säilyttämisestä välimuistissa. Sovellukset voivat pyytää tältä oliolta merkkijonoja tarvittaessa.



#### 7.4.2 Ohjainlaitteiden yhteyskäytännöt

Kuten aikaisemmin on mainittu, tarkoitetaan ohjainlaitteella MHP-standardissa kaukosäädintä, näppäimistöä tai hiirtä. Koodin lukijalaite voi kommunikoida MHP-päätelaitteen kanssa käyttäen näppäimistön merkkien valmiita ohjauskoodeja.

*org.dvb.event.UserEventRepository*:lle on sovelluskehyksessä aliluokka *ReaderEventRepository*, joka on vastuussa *PrintAccess*-koodien vastaanottamisesta ja tarkistamisesta sovelluskehyksessä. Esimerkiksi sovelluskehysten *CodeReader*-käyttöliittymäkomponentti käyttää jälkimmäistä koodien vastaanottamiseen. Päätelaite vastaanottaa näin ollen *PrintAccess*-koodin ikään kuin kaukosäätimen komentoina, mutta koodin prosessointi tapahtuu *CodeReader*- ja *ReaderEventRepository*-olioissa.

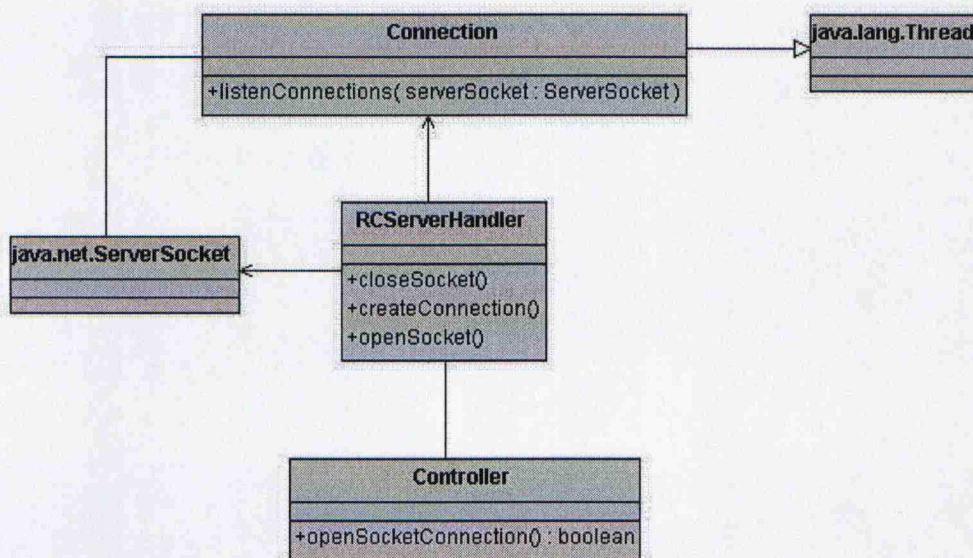
#### 7.4.3 TCP/IP –yhteydet MHP-päätelaitteelta

MHP-päätelaitteiden paluukanavaa kontrolloidaan *org.dvb.net.rc.RCInterfaceManager*-luokan avulla. Koska paluukanavaa voi päätelaitteessa käyttää vain yksi sovellus kerrallaan, lupa paluukanavan käyttöön on pyydettävä tältä luokalta. Kyseinen luokka on singleton-toteutus, jolloin siitä voidaan luoda vain yksi instanssi /47/. Viittaus tähän instanssiin pyydetään *RCInterfaceManager.getInstance* –metodin avulla.

Kehys tarjoaa ylemmän tason metodit paluukanavan käyttöön. Sovelluskehysten avulla paluukanavayhteys on avattavissa *Controller*-luokan *connectToReturnChannel*-metodilla. Tälle metodille annetaan parametrina se IP-osoite, johon paluukanavalla halutaan ottaa yhteys. Yhteys on tämän jälkeen käytettävissä normaalisti *java.net* –paketin luokkien avulla. Esimerkiksi HTTP-yhteyksien luominen tapahtuu tällöin *HttpURLConnection*-luokan *openConnection*-metodilla.

#### 7.4.4 TCP/IP –yhteydet MHP-päätelaitteelle

Paluukanavaa käyttävien yhteyksien kuunteleminen tapahtuu siten, että jokaisella kerralla sovelluskehystä käynnistettäessä yritetään muodostaa myös *java.net.ServerSocket*-yhteys. Tällöin yhteyden muodostamisesta vastaava säie asetetaan silmukkaan, joka kuuntelee tiettyä porttia (esimerkiksi 6789). Paluukanavan Socket-rajapinnan käsittelystä vastaa *RCServerHandler*-luokka. Silmukan toteuttaminen tapahtuu siten, että *Connection* luokka perii *java.lang.Thread*:ista ja silmukka on tässä luokassa omassa säikeessään. Tilanne on esitetty seuraavassa kuvassa (Kuva 19).



**Kuva 19.** *Socket-yhteyksien muodostamisesta ja hallinnasta vastaavat luokat*

Mikäli päätelaitteen IP-osoitteeseen ja ko. porttiin otetaan yhteys, päätelaite vastaanottaa sille lähetetyn merkkijonon ja tallettaa tämän *CodeStorage*-oliioon. Ko. varasto-oliosta sovellukset voivat hakea merkkijonon tarvittaessa. Java-kieli sisältää valmiit työkalut Socket-ohjelmointiin. *Socket* -luokkaan liitetään *InputStream*- ja *OutputStream*-oliot, jolloin vastaketta voi käyttää datavirran lukemiseen ja kirjoittamiseen samaan tapaan kuin normaalissa tiedostonhallinnassa.

## 7.5 Sovelluskehiksen toteutus ja dokumentointi

PrintAccess-sovelluskehys toteutettiin Java-ohjelmointikielellä käyttäen Sun Microsystemsin JDK 1.1.8-version mukaista luokkakirjastoa ja kääntäjää. MHP-luokkatiedostoista käytettiin version 1.0.2 mukaista referenssikirjastoa. Sovelluskehiksen ensimmäisen iteraation (versio 1.0) lähdekoodi on liitteessä 1.

Lähdekoodin dokumentointiin käytettiin JavaDoc-dokumentaatiomallin mukaista käytäntöä /11/. Siinä ohjelmakoodin sekaan kirjoitetaan luokkien ja metodien dokumentaatio standardien dokumentaatiomerkkien sisään. Standardista merkintätavasta on se hyöty, että dokumentaatiosta voidaan automaattisella Sun Microsystemsin tarjoamalla työkalulla generoida java-ohjelmoijille tuttu HTML-muotoinen dokumentaatio /11/. Sovelluskehiksen ja sen avulla laadittujen testikonseptien JavaDoc-käytännön mukainen dokumentaatio on liitteessä 2.



7.6 Sovelluskehityksen testaus

Kuluttajasovelluksiin tähtäävän MHP-sovelluskehityksen yksi suuri haaste on sovellusten luotettava testaaminen. Testaamiseen käytetään yleensä MHP:n referenssi-implementaatioita, eli päätelaitetta joka itsessään on läpäissyt standardin tietyn version mukaisen yhteensopivuustestin /42/. Sovelluskehittäjille tarkoitettulla referenssi-implementaatiolla ei kuitenkaan voida testata tiedostojen realistista yleislähetystä oliokarusellilla. Sovellusten lähetyksen ja vastaanottamisen testaaminen tuotantoympäristöä vastaavissa olosuhteissa edellyttääkin käytännössä tiivistä yhteistyötä digitaalisen television kanavaoperaattoreiden kanssa. /42/, /44/

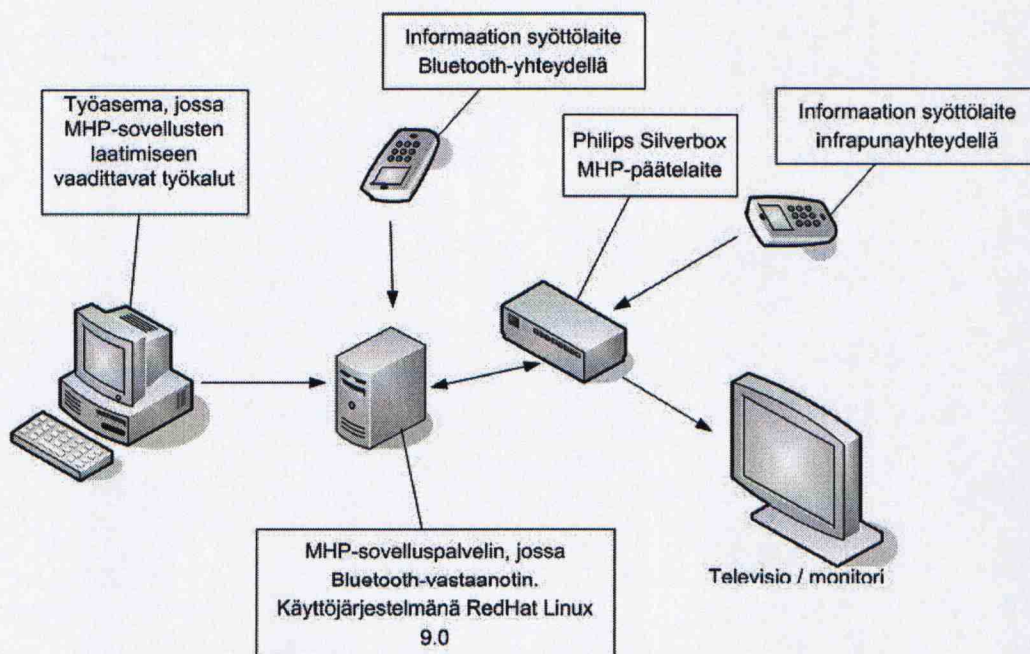
PrintAccess-sovelluskehityksen testaamiseksi TKK:n viestintätekniikan laboratorioon rakennettiin testausympäristö, jonka keskeisenä elementtinä on Philipsin MHP-sovelluskehitysalusta (Philips MHP SDK). Tässä ympäristössä MHP-päätelaitteena toimii Philips STMicro Silver-päätelaite. Kyseisen päätelaitteen tekniset ominaisuudet on esitetty alla olevassa taulukossa (Taulukko 7).

Taulukko 7. Testiympäristössä käytetyn MHP-päätelaitteen tekniset ominaisuudet /15/

| Ominaisuus                                | Arvo                    |
|---|-------------------------|
| Päätelaitteen tyyppi (prosessorin nopeus) | ST5512 (60 MHz)         |
| SDRAM-muisti                              | 16 MB                   |
| Flash-muisti                              | 8 MB                    |
| PC-korttipaikka DVB-CI-kortille           | Kyllä                   |
| SCART-liittimet                           | PAL- ja SECAM –tuki     |
| Yleislähetys                              | DVB-T ja DVB-S –tuki    |
| Sarjaportti                               | RS232 –liitäntä         |
| MHP-yhteensopivuus                        | Standardin versio 1.0.2 |

Testiympäristön päätelaitteeseen on mahdollista ladata MHP-sovelluksia käyttäen paluukanavaprotokollaa ja PPP (Point to Point Protocol) –yhteyttä. Tällöin päätelaite on yhdistetty sarjaportilla PC-koneeseen, jossa on PPP-palvelimen sisältävä Linux-käyttöjärjestelmä. Linux-koneessa ajetaan Apache-webpalvelinta, jossa MHP-sovelluksen käännetyt Java-binääritiedostot on asetettu tiettyyn oletushakemistoon. Kun MHP-päätelaite käynnistetään, se muodostaa yhteyden palvelinkoneeseen ja lataa sovellusten hallintaohjelmistonsa tiedot kaikista saatavilla olevista MHP-sovelluksista TCP/IP-protokollaa käyttäen. Tällainen menettely simuloi DVB-lähetysten oliokarusellia, jossa sovellusten binääritiedostot ja XML-muotoiset sisältötiedostot ladataan MHP-vastaanottimelle lähetysvirrasta.

Käyttäjä käynnistää valitsemansa sovellukset hallintaohjelmiston käyttöliittymästä päätelaitteen kaukosäätimellä, joka kommunikoi päätelaitteen kanssa infrapunayhteydellä. Alla olevassa kuvassa (Kuva 20) on esitetty testiympäristön toimintaperiaate.



**Kuva 20.** MHP-sovellusten testiympäristö TKK:n viestintätekniikan laboratoriossa

Sovelluskehityksen kehityksen aikaiseen testaamiseen käytettiin tapaa, jossa sovellukseen lisätään ajonaikaisesti poikkeusten käsittely- ja virheenkorjausinformaatiota, jonka perusteella voidaan päätellä sovelluksen tila. PrintAccess-sovelluskehys sisältää tähän valmiit työkalut. Päätelaitteen konsoliin tulostettavaa virheenkorjausinformaatiota on mahdollista seurata em. PPP-yhteyttä käyttäen Linuxin Telnet-ohjelmalla. Ohjelmakoodin testaaminen testiympäristön Philips Silverbox päätelaitteella oli kuitenkin melko hidasta, koska kunnollisia virheenkorjausmahdollisuuksia ei laitteessa ole ja useissa virhetilanteissa laite täytyykin käynnistää uudestaan, mikä vie aikaa laitteen ladatessa käyttöjärjestelmänsä aina uudelleen. Testaaminen oli työlästä sikäläkin, että kyseinen Philipsin testilaitteisto sisältää vielä paljon virheitä ja useita sovelluskehikseen vaadittuja toiminnallisuuksia jouduttiin testaamaan monilla erilaisilla tavoilla ennenkuin ne saatiin toimimaan.

PrintAccess-koodien syöttämiseen käytettiin testiympäristössä Nokia 6600- ja Nokia 3650-matkapuhelimia, joissa on kamera koodien lukemiseen painotuotteesta ja Bluetooth-yhteys koodien siirtämiseen MHP-päätelaitteelle. Koska testiympäristön MHP-päätelaitteessa ei ole Bluetooth-rajapintaa, simuloitiin sitä liittämällä erillinen USB Bluetooth-laite päätelaitteen kanssa kommunikoivaan Linux-koneeseen. Linux-koneessa ajettiin Bluetooth-protokollapinoa käyttävää sovellusta, joka ohjaa Bluetooth-laitteelta saamansa informaation suoraan MHP-päätelaitteen käyttämään vastakerajapintaan. Tällöin myös Bluetooth-yhteydellä lähetetty informaatio kulkee käyttäen paluukanavan protokollaa. Tilanne vastaa MHP-päätelaitetta, jossa paluukanavayhteydet olisi toteutettu Bluetooth-ominaisuuksilla varustetulla matkapuhelimella.



## 7.7 Testisovellus sovelluskehykselle

PrintAccess-projektissa laadittavien pilottisovellusten toteutus tapahtui eri vaiheissa. Vaihejako perustui pilottisovelluksiin toteutettavien ominaisuuksien määrään siten, että ensimmäisessä vaiheessa sovellukset sisälsivät tärkeimmät toiminnallisuudet kuten koodien vastaanoton, ja yksinkertaisten käyttöliittymien rakentamisen ja näyttämisen käyttäjälle.

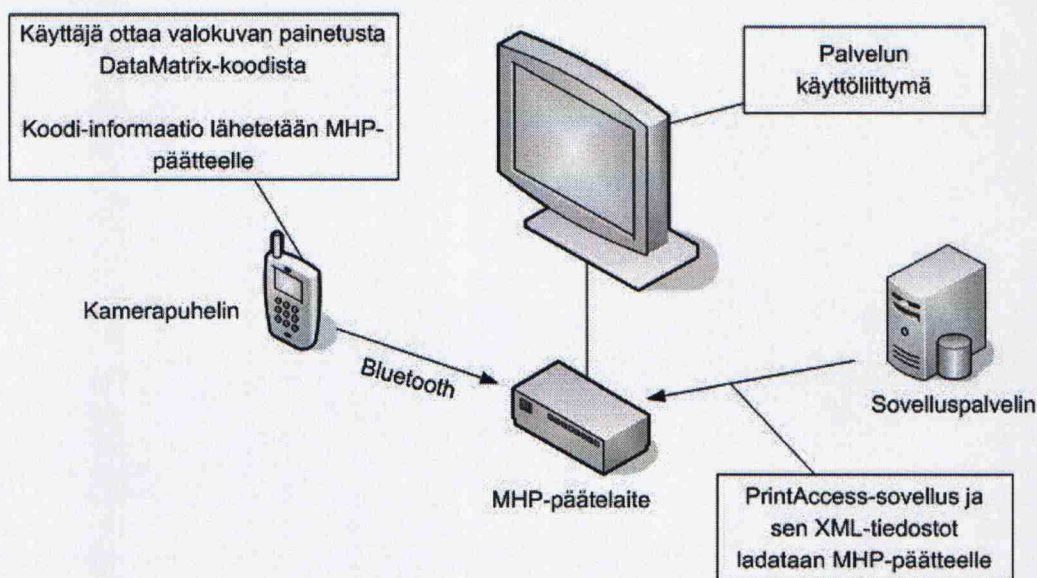
Kevään 2004 aikana projektissa toteutettiin kuusi ensimmäisen vaiheen testisovellusta, joista yksi toteutettiin MHP-alustalle käyttäen tässä diplomityössä kehitettyä PrintAccess-sovelluskehystä. Tämän sovelluksen toiminta-ajatuksena on tarjota digitaalisen television katselijalle reaaliaikaista informaatiota niistä urheilutapahtumista, jotka katsoja itse valitsee painotuotteesta kiinnostuksensa mukaan käyttämällä PrintAccess-koodeja. Sovellukselle annettiin projektissa nimeksi Score.

Sovelluksen toimintaperiaate on esitetty alla olevassa kuvassa (Kuva 21). Katsojaa kiinnostavien tapahtumien valinta suoritetaan sovelluksessa siten, että katsoja lukee painotuotteesta tietyn tapahtuman esittelyn kohdalle painetun kaksiulotteisen DataMatrix-viivakoodin ottamalla siitä kuvan matkapuhelimen kameralla. Kuvassa oleva koodi voidaan tulkita joko heti matkapuhelimessa tai vaihtoehtoisesti lähettämällä kuva MHP-päätelaitteeseen integroidulle koodintulkintaohjelmistolle, joka tulkitsee koodin ja siirtää sen sisällön MHP-ohjelmaan. Score-sovelluksen tapauksessa päädyttiin tukemaan molempia vaihtoehtoja, sillä mobiililla lukulaitteella (matkapuhelimella) suoritettava koodin tulkintaohjelmisto oli vielä keskeneräinen sovelluksen laatimista aloitettaessa ja koodin tulkinnan mahdollistava Java-ohjelma oli käytössä palvelinlaitteistolla.

Score-sovelluksen tapauksessa painotuotteisiin painettujen koodien sisältämä informaatio on URL-osoite, joka identifioi tietyn tapahtuman URL-osoitteen domain-nimiosuuden jälkeen asetetuilla parametreilla. Nämä parametrit ovat avaimia tietyn ottelun sisältötietoihin. Käytettävät URL-osoitteet ovat esimerkiksi muotoa:

[http://www.media.hut.fi/score?id=SM\\_498](http://www.media.hut.fi/score?id=SM_498), jossa id-parametri on ottelukohtainen tunnus.

Koodista tulkittu URL-osoite siirretään PrintAccess-sovelluskehiksen päälle toteutetulle MHP-sovellukselle käyttämällä matkapuhelimen Bluetooth-yhteyttä. Bluetooth-yhteyskäytäntö toteutettiin Score-sovelluksessa edellisessä luvussa esitellyllä TKK:n viestintätekniikan laboratorion testiympäristöllä.



**Kuva 21.** *PrintAccess-projektissa laaditun Score-sovelluksen toimintaperiaate*

URL-osoitetta vastaavasti MHP-päätelaite hakee lisäinformaatiota pelattavista otteluista tähän tarkoitukseen soveltuvalta J2EE-standardin kanssa yhteensopivalta avoimen lähdekoodin Tomcat-sovelluspalvelimelta. Palvelimella on servlet-teknologiaan perustuva yksinkertainen sovellus, joka kuuntelee yhteyspyyntöjä, lukee osoitteesta ottelua vastaavan ottelutunnuksen, tekee tarvittaessa pyyntöjä vastaavia hakuja ottelutietokantaan ja palauttaa ajankohtaisen informaation MHP-päätelaitteelle XML-tiedostona. Testisovelluksessa käytettiin PrintAccess-sovelluskehityksen valmiita tietoliikennearajapintoja, XML-sisällön tulkitsemiseen soveltuvia luokkia ja käyttöliittymäkomponentteja.

MHP-päätelaite saa jääkiekko-ottelun tulostiedot DVB-lähetysvirrasta oliokarusellin XML-sisältötiedostoina. Sisältötiedostot sisältävät tunnuksen (match\_id), joka toimii avaimena tietyn ottelun tietoihin. Edellä esitetyn URL-osoitteen avain (id) oli siten SM\_498. Vastaavalla tavalla koko XML-tiedosto voidaan nimetä käyttäen tätä tunnusta (esim. SM\_498.xml). Tällöin mikäli lähetysvirrassa on useiden ottelujen tiedot, hakee sovellus oikean ottelun tiedot tiedostonimen perusteella. Ottelutiedot päivitetään asettamalla ottelun uudet tiedot sisältävä XML-tiedosto lähetysvirtaan. Periaate on vastaava Suomen super-teksti-televisiossa käytössä olevissa sisältötiedostoissa. Esimerkki Score-ottelutietopalvelun XML-sisällöstä on esitetty alla (kuva 22).



```

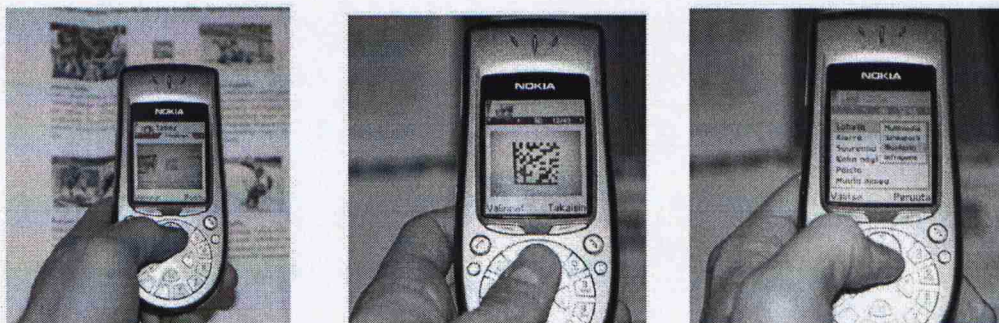
<score_data>
  <match_id="SM_498">
    <home_team>Kärpät</home_team>
    <visitor_team>Blues</visitor_team>
    <info_update_date>1082714742218</info_update_date>
    <match_status>ongoing</match_status>
    <match_time>25:25</match_time>
    <situation_in_game>1-3</situation_in_game>
    <goals>
      <goal>
        <score>1-0</score>
        <time>08:25</time>
        <scorer>Vesa Maijanen</scorer>
      </goal>
      <goal>
        <score>1-1</score>
        <time>12:18</time>
        <scorer>Pertti Lehtonen</scorer>
      </goal>
      <goal>
        <score>1-2</score>
        <time>18:30</time>
        <scorer>Mika Myllyjoki</scorer>
      </goal>
      <goal>
        <score>1-3</score>
        <time>24:20</time>
        <scorer>Perttu Mäkiäho</scorer>
      </goal>
    </goals>
  </match>
</score_data>

```

**Kuva 22.** Ote Score-ottelutietopalvelun xml-sisällöstä

Sovelluksessa käytettävät ottelutunnukset sijaitsevat omassa XML-tiedostossaan. Tiedosto on erillinen, koska siinä missä ottelutulosten sisältötiedot muuttuvat usein, tarvitsee koodit sisältävä XML-tiedosto ladata MHP-päätelaitteelle vain kerran. Kun päätelaite tietää käytettävät ottelutunnukset etukäteen, voidaan niiden oikeellisuudesta varmistua jo siinä vaiheessa kun PrintAccess-koodi lähetetään päätelaitteelle.

Seuraavissa kuvissa (kuvat 23 – 25) on esitetty Score-sovelluksen todellinen käyttötilanne. Painotuotteeseen on painettu tietoja jääkiekko-otteluista ja jokaisen ottelun tietojen viereen PrintAccess-koodi, joka sisältää avaimen kyseisen ottelun tietoihin Score-sovelluksessa. Käyttäjä lukee koodin Nokia 3650-matkapuhelimeen integroidulla kameralla ja siirtää kuvainformaation eteenpäin Bluetooth-yhteydellä (Kuva 23).



**Kuva 23.** *PrintAccess-koodi luetaan matkapuhelimen kameralla ja lähetetään Bluetooth-yhteydellä eteenpäin.*

Television MHP-vastaanottimella käynnistetään ottelutietopalvelu, joka toimii PrintAccess-sovelluskehityksen päällä. Sovellus jää käynnistämisen jälkeen odottamaan sille tietoliikenneporttiin lähetettävää PrintAccess-koodia (Kuva 24).



**Kuva 24.** *Painotuotteesta luettu PrintAccess-koodi lähetetään DVB-MHP-päätelaitteelle, jossa suoritetaan PrintAccess-sovelluskehityksen päälle rakennettua Score-sovellusta.*

Painotuotteesta luettu PrintAccess-koodi siirretään Bluetooth-yhteydellä MHP-sovellukselle, joka lukee koodista ottelukohtaisen tunnuksen ja etsii lähetysvirran oliokarusellista tätä tunnusta vastaavan ottelun sisältötiedoston. Kyseisestä sisältötiedostosta luetaan ottelun ajankohtaiset tiedot, jotka näytetään katsojalle. Halutessaan katsoja voi saada tilannetietoa tarkempaa tietoa esimerkiksi ottelun maalintekijöistä (Kuva 25).





**Kuva 25.** DVB-lähetysvirrassa olevista XML-sisältötiedostoista esitetään käyttäjälle luettua PrintAccess-koodia vastaava informaatio. Tässä tapauksessa käyttäjä etsi tietoa Kärpät-Blues -jääkiekko-ottelusta.

## 7.8 Sovelluskehiksen hyödyntäminen testisovelluksessa

Arvio diplomityössä suunnitellun ja toteutetun PrintAccess-sovelluskehiksen käyttökelpoisuudesta ja siten myös onnistumisesta perustuu sille, kuinka paljon siitä on hyötyä Score-sovelluksen kaltaisten PrintAccess-sovellusten rakentamisessa. Kuten aikaisemmin todettiin, on laadukkaiden sovelluskehysten laatiminen iteratiivinen prosessi, jossa sovelluskehiksen ominaisuuksia lisätään ja muutetaan sitä mukaa kun tietoa sen avulla kehitetyistä sovelluksista saadaan kerätyksi. Nyt tehtävä arvio perustuu liitteessä 1 olevaan PrintAccess-sovelluskehiksen versioon 1.0. Tämän version lähdekoodiin on liitetty lisäksi myös edellä esitellyn Score-sovelluksen koodi, jonka liittäminen sovelluskehikseen toteuttaa kuvassa 18 esitetyn mallin sovellusten (konseptien) käynnistämisestä sovelluskehiksen kautta.

Taulukossa 6 esiteltiin kaikille PrintAccess-sovelluksille primääriset ja sekundääriset ominaisuudet. Primäärisillä ominaisuuksilla tarkoitetaan kaikille alueen sovelluksille välttämättömiä ominaisuuksia. Ne toimivat siten motiivina sovelluskehiksen rakentamiselle. PrintAccess-sovelluskehys tarjoaa kaikki primääriset ominaisuudet sovellusten käyttöön ja Score-sovelluksessa hyödynnettiin kaikkia primäärisiä ominaisuuksia lukuunottamatta taulukon 6 ominaisuutta 7 (johtuen sovelluskehiksellä toteutettujen sovellusten vähäisestä lukumäärästä). Oleellisia Score-sovelluksessa käytettyjä sovelluskehiksen palveluja ovat vastakehteyden hyödyntäminen PrintAccess-koodien lähettämisessä ja XML-muotoisen ottelusisällön tulkitseminen. Myös käyttöliittymän valmiita komponentteja hyödynnettiin. Sovelluskehiksen nykyinen versio tarjoaa kuitenkin käyttöliittymien tekemiseen melko suppean valikoiman komponentteja, joten sovelluksia rakennettaessa on useimmiten luotava uusia komponentteja, jotta saataisiin aikaan halutun näköinen käyttöliittymä.



## 8 JOHTOPÄÄTÖKSET JA YHTEENVETO

Hybridimediateknologioilla pyritään kaventamaan eri viestintävälineiden välisiä rajoja niin, että eri medioiden yhteiskäyttö lisääntyisi ja käyttäjä voisi siirtyä joustavasti mediavälineestä toiseen hyödyntäen kunkin median ominaisuudet parhaiten käyttötilanteeseen sopivalla tavalla. Tämä diplomityö tehtiin osana TKK viestintätekniikan ja VTT median yhteistä PrintAccess-projektia, jossa tutkitaan ja kehitetään erilaisia painetun ja digitaalisen median yhteiskäyttösovelluksia.

Digitaalisen television ohjelmointiympäristöt kuten Multimedia Home Platform (MHP) mahdollistavat interaktiivisten sovellusten laatimisen digi-tv-vastaanottimiin. Diplomityössä selvitettiin edellytykset tällaisten sovellusten tekemiseen PrintAccess-tarkoituksiin ja toteutettiin MHP-standardin mukainen sovelluskehys, joka tarjoaa ohjelmistokehittäjille valmiit rajapinnat PrintAccess-sovellusten kehittämiseen. Näiden rajapintojen ominaisuudet määritettiin tarkastelemalla MHP-ympäristön ja PrintAccess-käyttötilanteiden erikoisominaisuuksia. Sovelluskehys toteutettiin ja testattiin MHP-standardin version 1.0.2 mukaisesti TKK viestintätekniikan laboratorion ohjelmointiympäristössä. Käytännön testit sovelluskehykselle tehtiin toteuttamalla kehyksen avulla yksi esimerkki PrintAccess-sovelluksesta, jossa käyttäjä pystyi vastaanottamaan televisio-ohjelmaan jääkiekko-otteluiden tilannetietoja painotuotteesta luettua ottelutunnusta vastaavasti.

Tärkeimpänä PrintAccess-sovellusten asettamana edellytyksenä on mahdollisuus TCP/IP-yhteyksien ja laitekohtaisten kaukosäädintoteutusten hyödyntämiseen. Näissä sovelluksissa PrintAccess-koodin lukulaite digitalisoi painotuotteesta olevan viestin ja lähettää sen MHP-päätelaitteelle digitaalisessa muodossa käyttäen edellämainittuja tietoliikenne-rajapintoja. PrintAccess-sovelluskehyksessä keskityttiin käyttämään MHP-standardin mukaisen paluukanavayhteyden TCP/IP-protokollaa ja vastakehyttä painotuotteesta luetun informaation siirtämiseksi PrintAccess-koodin lukijalaitteelta MHP-alustalle. Koska nykyisissä markkinoilla olevissa MHP-toteutuksissa paluukanavaratkaisuissa käytetään lähinnä puhelinmodeemia, käytettiin PrintAccess-sovellusten kehityksessä Linux-käyttöjärjestelmän päälle rakennettua kevyttä välityspalvelinta, joka välittää matkapuhelimelta esimerkiksi Bluetooth-yhteyden kautta saadun informaation eteenpäin päätelaitteelle käyttäen sen omaa paluukanavaa.

MHP-standardin tulevilla versioilla on todennäköisesti tuki jollekin tai joillekin erityisesti viihdelaitteiden käyttötarkoituksiin suunnitelluille sovellustason tietoliikenneprotokollille kuten esimerkiksi OSGi:lle. Lisäksi päätelaitteiden paluukanavavalikoima varmasti paranee sisältäen lähitulevaisuudessa ainakin ethernet-liitännäisen paluukanavan. Tällaisen kehityksen myötä kodin viihdelaitteiden välinen kommunikointi yleistyy ja vakiintuu, jolloin matkapuhelimella voi ohjata viihdelaitteita esimerkiksi juuri PrintAccess-koodien avulla.

Diplomityössä kehitetyn PrintAccess-sovelluskehysten kaltainen malli tulee MHP-päätelaitteissa todennäköisesti yleistymään tallennuskapasiteetin lisäyksen myötä. Tällöin sovel-



luskehyksen luokkatiedostoja voidaan ladata päätelaitteelle pysyvästi, jolloin digitaalisen television oliokaruselliin ei tarvitse laittaa kuin tietyn sovelluksen vaihtuvat tiedostot. Tämä säästää lähetysskapasiteettia ja nopeuttaa sovellusten lataamista päätelaitteilla.

Yksi MHP-ympäristön ominaisuuksista on, että se mahdollistaa ohjelmiin sidotut palvelut. Tulevaisuudessa kuitenkin, MHP:n Internet-profiiliin myötä, myös ohjelmista riippumattomat IP-pohjaiset palvelut kuten www-selailu ja sähköposti tulevat todennäköisesti olemaan osa televisioympäristöä. Tämä asettaa monia uusia mahdollisuuksia myös erilaisille PrintAccess-konsepteille. IP-yhteyksien kehittyessä PrintAccess-sovellusten käyttöön saadaan laajemmat XML- tietosisällöt ja tietokantayhteydet.

MHP-standardi on vielä kehityksensä alkuvaiheessa, mistä syystä nykyiset MHP-päätelaitteet ja niille laaditut sovellukset ovat teknisiltä ominaisuuksiltaan toistaiseksi varsin rajallisia. Mikäli MHP-standardi yleistyy Euroopassa ja muualla maailmassa, tulee siitä laitevalmistajille vähitellen yhä houkuttelevampi tuotekehityskohde ja markkinoille saadaan uusia monipuolisempia päätelaitteita. PrintAccess-sovellusten kannalta on erityisen tärkeää, että laitteiden kommunikaatorajapinnat kehittyvät ja markkinoille tulee päätelaitteita, jotka parantavat digi-tv-päätelaitteen toimintaa osana kodin langallisia ja langattomia lähiverkkoja. Tällainen kehityskulku vaikuttaa erittäin todennäköiseltä kun tarkastellaan nykyistä kodin viihdelaitteiden kehitystä kohti monipuolisesti varustettuja viihdekeskuksia. MHP-standardi ei ota kantaa tietoliikennesyhteyksien fyysisen tason toteutuksiin. Tällöin on laitevalmistajista riippuva seikka mitä paluukanavarajapintaa päätelaite tukee. PrintAccess-sovellusten näkökulmasta lupaavimpia ja käyttökelpoisimpia yhteystapoja ovat Bluetooth-yhteys ja WLAN-teknologia. Bluetooth on sisäänrakennettuna jo useisiin uusiin matkapuhelimiin ja WLAN on yleistynyt viime aikoina kodin lähiverkon toteutustekniikkana. Näiden tekniikoiden avulla lukulaitteiden avulla painotuotteesta luetut koodit ovat helposti siirrettävissä alustojen välillä.

Olemassaolevien MHP-sovellusten pienen lukumäärän vuoksi myös PrintAccess-sovelluskehitykselle asetettuja vaatimuksia on vaikea kartoittaa. Tästä syystä PrintAccess-projektin jatkotutkimuksen myötä sovelluksille asetettavat vaatimukset kasvavat ja tarkentuvat, jolloin myös sovelluskehityksestä on tarpeen luoda uusia versioita. Tarkka projektissa toteutettavien konseptien kartoitus ja varsinainen konseptitestausta jää näin myöhemmän tutkimuksen osaksi. PrintAccess-projektissa on myös yhtenä osa-alueena kehitetty yksi- ja kaksiulotteisten viivakoodien lukuohjelmistoja matkapuhelimissa käytettävälle Symbian-alustalle. Tällaisessa käyttötilanteessa käyttäjä joutuu hankkimaan lukuohjelmiston matkapuhelimeensa erikseen, jotta voisi käyttää PrintAccess-sovelluksia. Tästä saattaa aiheutua käyttäjälle ylimääräistä vaivaa mikä vaikuttaa kaikkien sovellusten käytettävyyteen. Tässä diplomityössä laadittuun sovelluskehitykseen liittyvänä jatkotutkimuksen kohteena voisi tämän perusteella olla PrintAccess-koodien, kuten esimerkiksi kaksiulotteisten DataMatrix-viivakoodien tulkitseminen vasta MHP-päätelaitteella. Tässä tapauksessa sovelluskehitykseen toteutettaisiin koodin tulkintaan suoraan kuvasta tarvittavat algoritmit. Kyseinen ominaisuus poistaisi tarpeen ladata mitään ylimääräistä ohjelmistoa matkapuhelimeen, jolloin käyttäjän kynnys PrintAccess-sovellusten käyttämiseen olisi varmasti matalampi.

## LÄHDELUETTELO

- /1/ **Allamaraju, S, Buest, C. & al.;** *Professional Java Server Programming J2EE 1.3 Edition*, Wrox Press 2001, 1189s.
- /2/ **Anon.** *ATK-Sanakirja*. Tietotekniikan liitto, Helsinki 2003. Saatavilla [www-muodossa](http://www.muodossa):  
<URL: <http://www.tt-tori.fi/atk-sanakirja/englanti-suomi/sisallys.htm>>, [viitattu 28.5.2003].
- /3/ **Anon.** *A Primer on Remote Control Technology*. Innotech Systems Inc, Yhdysvallat 2001, päivitetty 13.7.2001. Saatavilla [www-muodossa](http://www.muodossa):  
<URL:<http://www.innotechsystems.com/primer1.pdf>>, [viitattu 1.8.2003]
- /4/ **Anon.** *Digi-tv:n keskitaajuudet ja kanavanumerot*. Digita Oy, Helsinki 2004, päivitetty 19.4.2004. Saatavilla [www-muodossa](http://www.muodossa):  
<URL:<http://www.digitv.fi/sivu.asp?path=1;2999;3751>>, [viitattu 20.4.2004].
- /5/ **Anon.** *Digi-tv –kysymyksiä ja vastauksia*, Digita Oy, Helsinki 2003, päivitetty 9.1.2004. Saatavilla [www-muodossa](http://www.muodossa): <URL:  
<http://www.digitv.fi/sivu.asp?path=1;3000>>, [viitattu 12.1.2004]
- /6/ **Anon.** *Digisovittimien määrä suomalaisissa kotitalouksissa 15.10.2003*. Tiedote, Finnpanel Oy 15.10.2003. Saatavilla [www-muodossa](http://www.muodossa): <URL:  
[http://www.finnpanel.fi/tulokset/tiedotteet/tv\\_151003.html](http://www.finnpanel.fi/tulokset/tiedotteet/tv_151003.html)>, [viitattu 16.11.2003]
- /7/ **Anon.** *Digivastaanottimien määrä Suomessa*. Finnpanel Oy, Helsinki 2003. Lehistötiedote, päivitetty 15.10.2003. Saatavilla [www-muodossa](http://www.muodossa):  
<URL:[http://www.finnpanel.fi/tulokset/tiedotteet/tv\\_151003.html](http://www.finnpanel.fi/tulokset/tiedotteet/tv_151003.html)>, [viitattu 18.10.2003].
- /8/ **Anon.** *European Broadcasting Union, Multimedia Home Platform 1.0.2 – DVB Bluebook A057 Rev.2*, 765 s. Saatavilla [www-muodossa](http://www.muodossa): <URL:  
<http://www.mhp.org>>, [viitattu 13.6.2003]
- /9/ **Anon.** *Infrared remote control technology. IR remote control. General info*. ELH Communications Ltd 2001. Saatavilla [www-muodossa](http://www.muodossa):  
<URL:<http://links.epanorama.net/links/irremote.html>>, [viitattu 1.12.2003]
- /10/ **Anon.** *IPDC Content Services White Paper*, Sonera MediaLab, Suomi 2003, 21 s. Saatavilla [www-muodossa](http://www.muodossa): <URL: <http://ipdc-forum.org/about/documents/IPDCContentServicesWP.pdf>>, [viitattu 2.12.2003]



- /11/ **Anon.** *Javadoc – The Java API Documentation Generator*. Sun Microsystems 2001, saatavilla [www-muodossa: <URL: http://java.sun.com/j2se/1.3/docs/tooldocs/win32/javadoc.html>](http://java.sun.com/j2se/1.3/docs/tooldocs/win32/javadoc.html), [viitattu 4.11.2003]
- /12/ **Anon.** *Java Technologies for Interactive Television*. Technical White Paper Sun Microsystems 2001, saatavilla [www-muodossa: <URL: http://java.sun.com/products/javatv/whitepapers/TechInterTV052101.pdf>](http://java.sun.com/products/javatv/whitepapers/TechInterTV052101.pdf), [viitattu 12.6.2003]
- /13/ **Anon.** *Kohti yksilöllistä mediamaisemaa*. TEKES:in Kuluttajatutkimukset-hanke, Teknologia katsaus 98/2000, TEKES. 127 s.
- /14/ **Anon.** *Luvasta vapautetut radiolaitteet – LVR 1/2004*. Viestintävirasto, Helsinki 2004. Saatavilla [www-muodossa: <URL: http://www.ficora.fi/suomi/radio/lvr.htm>](http://www.ficora.fi/suomi/radio/lvr.htm), [viitattu 3.1.2004]
- /15/ **Anon.** *MHP SDK Installation & User Guide*, Philips DN Softworks, v. 1.5 10/2002. 44 s.
- /16/ **Anon.** *Mobile phone controls TV's*, Cellular News, 9.5.2003 saatavilla [www-muodossa: <URL: http://www.cellular-news.com/story/8830.shtml>](http://www.cellular-news.com/story/8830.shtml), [viitattu 3.8.2003]
- /17/ **Anon.** *Nokia 230T Mediamaster User Guide*. Nokia Oyj Espoo 2003, Saatavilla [www-muodossa: <URL: http://nds1.nokia.com/home\\_products/files/users\\_guide/230T\\_usersguide\\_en.pdf>](http://nds1.nokia.com/home_products/files/users_guide/230T_usersguide_en.pdf), [viitattu 10.12.2003]
- /18/ **Anon.** *PrintAccess-projektin tutkimussuunnitelma*. TKK Viestintätekniikka, VTT Media, 2003
- /19/ **Anon.** *PrintAccess user stories*, TKK Viestintätekniikka, VTT Media, 2003 6 s.
- /20/ **Anon.** *Specification of the Bluetooth System*, Bluetooth Organization 22.2.2001. Saatavilla [www-muodossa: <URL: https://www.bluetooth.org/foundry/specification/document/Bluetooth\\_Core\\_1.1\\_vol\\_1/en/1/Bluetooth\\_Core\\_1.1\\_vol\\_1.zip>](https://www.bluetooth.org/foundry/specification/document/Bluetooth_Core_1.1_vol_1/en/1/Bluetooth_Core_1.1_vol_1.zip), [viitattu 13.8.2003]
- /21/ **Anon.** *UML Notation Guide*, Rational Software 1.9.1997, versio 1.1. Saatavilla [www-muodossa: <URL: http://umlcenter.visual-paradigm.com/umlresources/nota\\_11.pdf>](http://umlcenter.visual-paradigm.com/umlresources/nota_11.pdf), [viitattu 13.8.2003]
- /22/ **Anon.** *XHTML 1.0 The Extensible Hypertext Markup Language*, W3C-konsortio 1.8.2002. Saatavilla [www-muodossa: <URL: http://www.w3.org/TR/xhtml1/>](http://www.w3.org/TR/xhtml1/), [viitattu 7.12.2003]

- /23/ **Antikainen, H., Siivonen, T.** *Viestintäalan nykytila ja kehitystrendit 2001-2002*, GT-raportti nro 1. Tammikuu 2001, 44 s.
- /24/ **Bencomo, N., Losavio, F., Marchena, F., Matteo, A.** *Java implementations of user-interface frameworks*, TOOLS 23 (Technology of Object-Oriented Languages and Systems) Proceedings, 1997. S. 232 –246.
- /25/ **Benoit, H.** *Digital Television, MPEG-1, MPEG-2 and principles of the DVB system*. Focal Press, Englanti 2002. 201 s.
- /26/ **Bonin J, Courtney J, Piesing J**, *MHP for Application Developers*, Presentation Notes at JavaOne Conference 2002, saatavilla [www-muodossa](http://www.muodossa.com/URL:http://servlet.java.sun.com/javaone/resources/content/sf2002/conf/sessions/pdfs/1521.pdf): URL: <http://servlet.java.sun.com/javaone/resources/content/sf2002/conf/sessions/pdfs/1521.pdf>], [viitattu 1.8.2003]
- /27/ **Bosch, J.** *Design & Use of Software Architectures*. Addison-Wesley, USA 2000. 354 s.
- /28/ **Bosch, J., Molin, P., Mattsson, M., Bengtsson, P.** *Object-oriented framework-based software development: problems and experiences*. ACM Computing Surveys (CSUR) 1/2000.
- /29/ **Carey, J., Carlson, B.** *Framework Process Patterns – Lessons Learned Developing Application Frameworks*, Addison-Wesley, USA 2002. 231 s.
- /30/ **Cesar, P.** *HAVi Components in Digital Television*, Diplomityö, TKK 2001, 73 s.
- /31/ **Chengyan, P.** *Digital Television Applications*, Väitöskirja, TKK 2002, 117 s.
- /32/ **Della, L., Clark, D.** *From interface to persistence: a framework for business oriented applications*. Technology of Object-Oriented Languages TOOLS 28. Proceedings 23-26 11/1998. s. 27 -39
- /33/ **De Scheemaeker, M.** *NanoXML/Java 2.2*, päivitetty 1.2.2003. Saatavilla [www-muodossa](http://www.muodossa.com/URL:http://web.wanadoo.be/cyberelf/nanoxml/downloads/NanoXML-Java.pdf) <URL: <http://web.wanadoo.be/cyberelf/nanoxml/downloads/NanoXML-Java.pdf> >, [viitattu 14.7.2003]
- /34/ **Ero, L., Jaakola, P., Kosonen, I., Miettinen, J., Ojala, S., Suomi, M., Tunturi, J.** *Televisiotoiminta Suomessa 2010*. Liikenne- ja viestintäministeriö, Helsinki 2003. Esiselvitys. 59 s. Saatavilla [www-muodossa](http://www.muodossa.com/URL:http://www.mintc.fi/lvm_old/data/www/sivut/dokumentit/julkaisu/julkaisusarja/2003/a302003.pdf): <URL: [http://www.mintc.fi/lvm\\_old/data/www/sivut/dokumentit/julkaisu/julkaisusarja/2003/a302003.pdf](http://www.mintc.fi/lvm_old/data/www/sivut/dokumentit/julkaisu/julkaisusarja/2003/a302003.pdf)>, [viitattu 28.9.2003]
- /35/ **ETSI TS 102 812 V1.1.1**, *Digital Video Broadcasting (DVB); Multimedia Home Platform (MHP) Specification 1.1*. European Telecommunications Standard Institute, 2001. 1447 s.



- /36/ **ETSI TS 101 812 V 1.2.1**, *Digital Video Broadcasting (DVB); Multimedia Home Platform (MHP) Specification 1.0.2*. European Telecommunications Standards Institute, 2002. 757s.
- /37/ **Falck, K.** *MHP – Interaktiivinen televisio*. Tietokone 9/2002. s 62-64.
- /38/ **Fayad, M., Johnson, R., Schmidt, D.** *Implementing Application Frameworks*. John Wiley & Sons, USA 1999. 729 s.
- /39/ **Fayad, M., Johnson, R., Schmidt, D.** *Building Application Frameworks*, John Wiley & Sons, USA 1999. 664 s.
- /40/ **Feldbusch, F., Paar, A., Odendahl, M., Ivanov, Ivan.** *The BTRC Bluetooth remote control system*. Personal and Ubiquitous Computing, vol 7, 2/2003, S. 102-112.
- /41/ **Fontoura M, Pree W, Rumpe B**, *The UML Profile for Framework Architectures*, Addison-Wesley 2001, 240 s.
- /42/ **Fotschl, H-P., Plosch, R.** *Interactive applications for the multimedia home platform*. Fourth International Symposium on Multimedia Software Engineering, Proceedings, 2002. S. 264 –271.
- /43/ **Fowler, M.** *Refactoring, Improving the Design of Existing Code*, Addison-Wesley, USA 1999. 431 s.
- /44/ **Frolich O**, *Establishment of MHP Application Development and play-out seen from a public broadcasters point of view*, saatavilla [www.muodossa](http://www.muodossa.com):  
<URL: <http://www.broadcastpapers.com/data/IBCDRMHPAppDevelopment.pdf>>, [viitattu 15.8.2003]
- /45/ **Gamma, E., Helm, R., Johnson, R., Vlissides, J.** *Design Patterns – Elements of Reusable Object-Oriented Software*. Addison Wesley, USA 2000. 386 s.
- /46/ **Gil, A.; Pazos, J.; Lopez, C.; Lopez, J.; Rubio, R.; Ramos, M.; Diaz, R.**, *Surfing the Web on TV: the MHP approach*, Multimedia and Expo, 2002. Proceedings. 2002 IEEE International Conference on , Volume: 2 , 26-29 Aug. 2002. S. 285 -288 vol.2
- /47/ **Grand, M.** *Patterns in Java*. Wiley & Sons, USA 1998. 467 s.
- /48/ **Granlund, K.** *Tietoliikenne*. Teknolit Oy, Jyväskylä 2000. 352 s.
- /49/ **Haataja, J.** *Digitaalisen television luettavuus*, Diplomityö TTK 2002. 129 s.

- /50/ **Hannula-Stenqvist, K.** *Digitaalinen televisio aloittaa 27.8.2001 – mikä on interaktiivisuuden rooli?.* Taideteollisen korkeakoulun lopputyö TAIK 2001. 100 s. Saatavilla [www-muodossa](http://www.muodossa.com): <URL: <http://www.mlab.uiah.fi/5medialaunch/lopputyokhs.pdf>>, [viitattu 18.6.2003]
- /51/ **HAVi Inc.** *The HAVi Specification – Specification of the Home Audio/Video Interoperability (HAVi) Architecture.* Versio 1.1, 529 s. Saatavilla [www-muodossa](http://www.muodossa.com): <URL: <http://www.havi.org>> [viitattu 13.6.2003]
- /52/ **Hiidenkari, R.** *Viestinten synergian kokeminen.* Diplomityö, TKK 2003, 128 s.
- /53/ **Hughes, M., Shoffner, M., Hamner, D.** *Java Network Programming.* Manning Publications. USA 1999. 807 s.
- /54/ **Iinatti, J., Latva-Aho, M.** *Tulevaisuuden radiorajapintaratkaisut: Kohti suurempaa tiedonsiirtonopeutta.* Prosempi 11/2002, Helsinki. S. 68-71.
- /55/ **ISO/IEC 16022.** *Information technology – International symbology specification – Data matrix.* International Organization for Standardization, 2000. 104 s.
- /56/ **Isohella, A.-R., Vauhti** koitui digi-tv:n turmioksi, Helsingin Sanomat 20.5.2003, s: B7.
- /57/ **Jones, J.** *DVB-MHP/Java TV Data Transport Mechanisms.* TOOLS 40 (Technology of Object-Oriented Languages and Systems) Proceedings, 1999. s. 28 – 39.
- /58/ **Juhola, H., Bäck, A., Siivonen, T., Lindberg, T., Pitkänen, M., Södergård, C., Nurmi, O.** *Digitaalinen painaminen osana tuotantoketjua,* VTT Tietotekniikka 1998, 116 s., Saatavilla [www-muodossa](http://www.muodossa.com): <URL: <http://www.vtt.fi/tte/samba/projects/digital/fin/pdf/loppuraportti.pdf>>, [viitattu 20.7.2003]
- /59/ **Kantola, K., Lahti, M., Väättänen, A.** *Ensiaskleet digi-tv:n katsojaksi.* Digitaalisen television käyttöönottokokeilu Pirkanmaalla. VTT Tiedotteita 2188, VTT Espoo 2003, 53 s.
- /60/ **Kardach, J.** *Bluetooth Architecture Overview,* Mobile Computing Group, Intel Corporation, Intel Technology Journal 2000, saatavilla [www-muodossa](http://www.muodossa.com): <URL: [http://www.intel.com/technology/itj/q22000/pdf/art\\_1.pdf](http://www.intel.com/technology/itj/q22000/pdf/art_1.pdf)>, [viitattu 18.7.2003]
- /61/ **Keinonen, T., Mattelmäki, T., Soosalu, M., Säde, S.** *Usability Design Methods.* Työraportti 6, KATTI-hanke. Taideteollinen Korkeakoulu 1997. 39 s. Saatavilla [www-muodossa](http://www.muodossa.com): <URL: <http://www.vtt.fi/tuo/45/projektit/katti/docs/udmtaik.pdf>>, [viitattu 15.12.2003]



- /62/ **Klemmer, S., Graham, J., Wolff, G., Landay, J.** *Books with voices: paper transcripts as a physical interface to oral histories*. Proceedings of the conference on Human factors in computing systems, USA 2003. S. 89-96.
- /63/ **Knutson, C., Folsom, B.** *IrMC: Infrared Solutions for Mobile Communications*, Counterpoint Systems Foundry Inc, USA. Saatavilla [www-muodossa](http://www.muodossa.com): <URL: [http://www.irda.org/design/irmc\\_solutions.pdf](http://www.irda.org/design/irmc_solutions.pdf)>, [viitattu 14.7.2003]
- /64/ **Knutson, C., Glade, D.** *Infrared Data Communications with IrDA*, Counterpoint Systems Foundry Inc, USA, Saatavilla [www-muodossa](http://www.muodossa.com): <URL: [http://www.irda.org/design/infrared\\_data\\_communications\\_with\\_irda.pdf](http://www.irda.org/design/infrared_data_communications_with_irda.pdf)>, [viitattu 14.7.2003]
- /65/ **Kojo, M.**, *A method to deliver multiple media content for digital television*, VTT Information Technology Research Report TTE4-2002-40, saatavilla [www-muodossa](http://www.muodossa.com): <URL: <http://www.vtt.fi/tte/mobtv/pub/kojo.pdf>>, [viitattu 1.8.2003]
- /66/ **Kontio, M.** *Mobiili Java J2ME*, Helsinki: Edita / IT Press 2002, 232 s.
- /67/ **Kunnas, J., Hagman, S., Kaasinen, E., Vanhanen, H.** *Integroitu julkaiseminen*, Digitaalisen median raportti 2/99, Espoo 1999, VTT. Saatavilla [www-muodossa](http://www.muodossa.com): <URL: [http://www.vtt.fi/imu/report/integroitu\\_julkaiseminen.htm](http://www.vtt.fi/imu/report/integroitu_julkaiseminen.htm)>, [viitattu 8.12.2003]
- /68/ **Lais, S.** *Optical Character Recognition*. ComputerWorld 7/2002. Saatavilla [www-muodossa](http://www.muodossa.com): <URL: <http://www.computerworld.com/softwaretopics/software/apps/story/0,10801,73023,00.html>>, [viitattu 13.12.2003]
- /69/ **Larman, G.** *Applying UML and patterns, An Introduction to Object-Oriented Analysis and Design and the Unified Process*, Prentice Hall, 2002 USA. 627 s.
- /70/ **Larsen, G.** *Designing component-based frameworks using patterns in the UML*, October 1999, Communications of the ACM, Volume 42 Issue 10
- /71/ **Lee, D.J.Y.; Basmaji, M.**; *Integrating mobile with IP phone*, Personal, Indoor and Mobile Radio Communications, 2002. The 13<sup>th</sup> IEEE International Symposium on, Volume 2, 15.9-18.9.2002, s. 982-986.
- /72/ **Lyytikäinen, K.**, *Sanomalehden asema nuorten aikuisten arjessa*. Pro-gradu tutkielma, Helsingin yliopisto 2003, 121 s.
- /73/ **Mahemoff, M.J., Johnston, L.J.** *Handling multiple domain objects with Model-View-Controller*. TOOLS 32 (Technology of Object-Oriented Languages and Systems) Proceedings, 1999. S. 28 –39.

- /74/ **Markiewicz, M., Lucena C.** *Object Oriented Framework Development*, July 2001 Crossroads, Volume 7 Issue 4 Espanja , saatavilla [www-muodossa](http://www.muodossa.com): <URL: <http://www.acm.org/crossroads/xrds7-4/frameworks.html>>, [viitattu 15.5.2003]
- /75/ **Mettälä, R.** *Bluetooth Protocol Architecture Version 1.0*, White paper, Nokia Oyj, 1999. Saatavilla [www-muodossa](http://www.muodossa.com): <URL:[https://www.bluetooth.org/foundry/sitecontent/document/Protocol\\_Architecture](https://www.bluetooth.org/foundry/sitecontent/document/Protocol_Architecture)> [viitattu: 18.7.2003]
- /76/ **Miletzki, U.**; *Character recognition in practice today and tomorrow*, Document analysis and Recognition, 1997. Proceedings of the Fourth International Conference on, Volume 2, 18-20 Aug 1997, Pages 902-907.
- /77/ **Millar I., Beale. M., Donoghue, B., Lindstrom, K., Williams, S.** *The IrDA Standards for High-Speed Infrared Communications*. The Hewlett-Packard Journal. Article 2/1998. 20s. Saatavilla [www-muodossa](http://www.muodossa.com): <URL: [http://www.irda.org/design/high\\_speed.pdf](http://www.irda.org/design/high_speed.pdf)>, [viitattu 13.12.2003]
- /78/ **Miller, B., Bisdikian C.** *Bluetooth Revealed*, Prentice Hall 2000, 320 s.
- /79/ **Morris, S.** *The MHP Tutorial*, The Interactive TV Web 2002, saatavilla [www-muodossa](http://www.muodossa.com): URL: <http://www.mhp-interactive.org/tutorial/mhp/index.shtml>>, [viitattu 14.6.2003]
- /80/ **MyTV-konsortio.** *MyTV - personalised services for digital television*. Projektin loppuraportti 2002. 23 s. Saatavilla [www-muodossa](http://www.muodossa.com): <URL: <http://www.extra.research.philips.com/euprojects/mytv/myTV%20Final%20project%20overview.pdf>> [viitattu 20.8.2003]
- /81/ **Nakhimovsky, A., Myers, T.** *Professional Java XML Programming*. Wrox Publishing USA 2000. 772 s.
- /82/ **Newmarch, J.** *Guide to JINI Technologies*, Version 3.02, 2003. Saatavilla [www-muodossa](http://www.muodossa.com): <URL: <http://pandonia.canberra.edu.au/java/jini/tutorial/Jini.xml>>, [viitattu 12.11.2003]
- /83/ **Niskanen, P.** *WAP – Wireless Application Protocol*, ITPress Helsinki 2000, 221 s.
- /84/ **Noppiari, E.**; *Julkinen tila viestintäympäristönä – Mobiilitelevision käyttökontekstia kartoittamassa*. Tampereen yliopisto, Journalismin tutkimusyksikkö, 2002. Saatavilla [www-muodossa](http://www.muodossa.com): URL: <http://www.vtt.fi/tte/mobtv/presentation.htm>>, [viitattu 3.8.2003]



- /85/ **Nordig**, *NorDig Requirements for the Basic TV profile for Digital Integrated Receiver Decoders for use in cable, satellite, terrestrial and IP-based networks*, NorDig 2003, saatavilla [www-muodossa](http://www.nordig.org/pdf/NorDig-Basic_ver_1.0.1.pdf): <URL: [http://www.nordig.org/pdf/NorDig-Basic\\_ver\\_1.0.1.pdf](http://www.nordig.org/pdf/NorDig-Basic_ver_1.0.1.pdf)> [viitattu 20.8.2003]
- /86/ **North, S., Hermans, P.** *XML, parhaat ratkaisut ja olennaiset taidot*, ITPress, Helsinki 2000, 514s.
- /87/ **Oesch, K., Varesmaa, A., Nummenpää, T., Vuorimaa, P.** *Verkostotalouden uudet sovellukset – Aihealueen tulevaisuuden suuntauksia ja kehittämistarpeita*, TEKES – Teknologia katsaus 136/2003, Saatavilla [www-muodossa](http://www.tekes.fi/julkaisut/verkostotalouden_uudet_sovellukset.pdf): <URL: [http://www.tekes.fi/julkaisut/verkostotalouden\\_uudet\\_sovellukset.pdf](http://www.tekes.fi/julkaisut/verkostotalouden_uudet_sovellukset.pdf)>, [viitattu 20.7.2003]
- /88/ **OSGi**, *OSGi Service Platform - specification release 3*, OSGi Alliance, 2003. 602 s.
- /89/ **Parviainen, J.** *Käyttäjän tunnistaminen digitaalisessa televisiossa*. Diplomityö, TKK, 2003. 59 s.
- /90/ **Pelkonen, T., Kalli, S.** *Digitaalisen television ansaintalogiikat: Toimintakenttä, liiketoimintamallit ja tulevaisuudennäkymät*, loppuraportti, Viestintäministeriö 2002, saatavilla [www-muodossa](http://www.mintc.fi/www/sivut/dokumentit/julkaisu/julkaisusarja/2002/a252002r.pdf): <URL: <http://www.mintc.fi/www/sivut/dokumentit/julkaisu/julkaisusarja/2002/a252002r.pdf>>. [viitattu 1.8.2003]
- /91/ **Piesing, J.;** *The DVB multimedia home platform - "MHP"*, IEE Colloquium on Interactive Television, 6.12.1999, s. 2/1 -2/6
- /92/ **Pääkkönen, P.** *Kodin verkotettujen laitteiden palveluiden hyödyntäminen*. VTT Elektronikka 2002, 76 s. Saatavilla [www-muodossa](http://www.inf.vtt.fi/pdf/tiedotteet/2002/T2157.pdf): <URL: <http://www.inf.vtt.fi/pdf/tiedotteet/2002/T2157.pdf>>, [viitattu 28.12.2003]
- /93/ **Rantala, K.** *Voisiko teräväpiirtotelevisio tulla Suomeen?* YLE Tekniikan tiedotuslehti, Huhtikuu 2002, Saatavilla [www-muodossa](http://www.yle.fi/tekniikka/tklehti/tk90/hdtv_suo.htm): <URL: [http://www.yle.fi/tekniikka/tklehti/tk90/hdtv\\_suo.htm](http://www.yle.fi/tekniikka/tklehti/tk90/hdtv_suo.htm)>, [viitattu 1.12.2003]
- /94/ **RFC 1060**, *RFC 1060 – Assigned numbers*. Network Working Group, 1990, Saatavilla [www-muodossa](http://www.faqs.org/rfcs/rfc1060.html): <URL: <http://www.faqs.org/rfcs/rfc1060.html>>, [viitattu 8.12.2003].
- /95/ **Rinnetmäki, M., Pöyhtäri A.** *Digi-TV:n palveluntekijän opas*, TEKES 2001, Saatavilla [www-muodossa](http://www.tekes.fi/julkaisut/Digi_TV_opas.pdf): <URL: [http://www.tekes.fi/julkaisut/Digi\\_TV\\_opas.pdf](http://www.tekes.fi/julkaisut/Digi_TV_opas.pdf)>, [viitattu 15.5.2003]
- /96/ **Risberg, K.** *DVB-multipleksi*, Ylen tekniikan tiedotuslehti, nro 64 joulukuu 1998. Saatavilla [www-muodossa](http://www.yle.fi/tekniikka/tklehti/tk64/dvb.htm): <URL: <http://www.yle.fi/tekniikka/tklehti/tk64/dvb.htm>>, [viitattu 18.5.2003]

- /97/ **Ritter, M.** *The Future of WLAN*. Queue vol. 1, 3/2003 Wireless Revolution, s. 18-27
- /98/ **Räihä, K-J., Jääskeläinen, K., Oesch K.** *Uusi käyttäjäkeskeinen tietotekniikka*. Digitaalisen median raportti 5/1999, Tekes. 71 s.
- /99/ **Saikanmäki, A., Nykänen, P.** *Digi-TV:n mobiilipalukanava*. Liikenne- ja viestintäministeriö, Helsinki 2002, Raportti 11 s. Saatavilla www-muodossa: <URL: <http://www.mona-ohjelma.net/julkaisut/DTV-raportti.pdf>>, [viitattu 10.12.2003]
- /100/ **Savia, O-P.** *DVB-vastaanottimen laitteistorajapinnan Java-toteutus*. TTKK, Tietotekniikan osasto, Tampere 1999. Diplomityö. 57 s.
- /101/ **Savolainen, S.** *Comparison of wireless short-range connectivity alternatives*. Diplomityö TKK Sähkö- ja tietoliikennetekniikan osasto, 2001 84 s.
- /102/ **Shirazi, J.** *Java Performance Tuning*, O'Reilly & Associates 2000, 426 s.
- /103/ **Soronen, H.**, *Tulevaisuuden sähköiset palvelut eri päätelaitteissa*. In: Digitalisoituvan viestinnän monet kasvot, Helsinki 2001, Tekes, S. 31-53. saatavilla www-muodossa: <URL: [http://www.tekes.fi/julkaisut/Digitalisoituvan\\_viestinnan\\_monet\\_kasvot.pdf](http://www.tekes.fi/julkaisut/Digitalisoituvan_viestinnan_monet_kasvot.pdf)>, [viitattu 6.6.2003]
- /104/ **Ström, O.** *Karuselliajeltua digitaalisilla teillä*, Ylen tekniikan tiedotuslehti Maaliskuu 2/2001, saatavilla www-muodossa: <URL: <http://www.yle.fi/tekniikka/tklehti/tk81/karu.htm>>, viitattu 20.7.2003
- /105/ **Södergård, C.** *Netti-televisio korvasi päivälehdet*, Prosessori 11/2003 s. 20-22.
- /106/ **Södergård, C., Ollikainen, V.** *Digitaalisten televisiolähetysten käyttö datajakelussa*, VTT Tietotekniikka 1999, 72 s. Saatavilla www-muodossa: <http://www.inf.vtt.fi/pdf/tiedotteet/1999/T1971.pdf>, viitattu 13.7.2003
- /107/ **Travis, G.** *Understanding the Java classloader*. IBM Developerworks, 2001, Saatavilla www-muodossa: <URL: <https://www6.software.ibm.com/developerworks/education/j-classloader/index.html>>, [viitattu 27.7.2003]
- /108/ **Vaxevanakis, K., Zahariadis, T., Vogiatzis, N.** *A review on wireless home network technologies*. Mobile computing and communications Review 2/2003, ACM Sigmobile. S. 59-68.
- /109/ **Verhoeven, R., Dees, W.** *Defining Services for Mobile Terminals using Remote User Interfaces*. Vivian Project Publications, Nokia 2002, saatavilla www-muodossa: <URL: <http://www-nrc.nokia.com/Vivian/Public/Misc/artVerhVR.pdf>>, [viitattu 18.8.2003]



- /110/ **Vuori M., Kivistö-Rahnasto J.** *Käyttöliittymien kehittämisen työkalupakki*. VTT Automaatio 2000, 140 s. Saatavilla [www-muodossa: <URL: http://www.vtt.fi/tuo/45/projektit/smart/docs/tk-doit.pdf>](http://www.vtt.fi/tuo/45/projektit/smart/docs/tk-doit.pdf), [viitattu 20.9.2003]
- /111/ **Vuorimaa, P., Cesar, P.** *A Graphical User Interface Framework for Digital Television*. Proceedings of the 10th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, WSCG'2002, Posters, 2002, S. 1-4.
- /112/ **Vuorimaa, P., Sancho, C.** *XML based text TV*. Proceedings of the First International Conference on Web Information Systems WISE 2000, Volume: 1, s. 109 –113.
- /113/ **Vogt, C.,** *The DVB MHP Specification – A Guided Tour*, World Broadcast Engineering – Spring edition 8 s. Saatavilla [www-muodossa: <URL: http://www.alticast.co.kr/downloads/MHP%20Guide.pdf>](http://www.alticast.co.kr/downloads/MHP%20Guide.pdf), [viitattu 20.8.2003]
- /114/ **Wootton, C.** *JavaScript Programmer's reference*. Wrox Publishing, USA 2001. 905 s.
- /115/ **Wu, C-H., Irwin, J.** *Emerging Multimedia Computer Communication Technologies*. Prentice Hall, USA 1998. 443 s.
- /116/ **Zhang, J., Chen, X., Yang, J., Waibel, A.** *A PDA-based sign translator*. Multimodal Interfaces, 2002. Proceedings. Fourth IEEE International Conference. s. 217 - 222

## LIITTEET

### **Liite 1: PrintAccess-sovelluskehityksen lähdekoodi**

<http://www.hut.fi/~tvenho/thesis/code/>

### **Liite 2: PrintAccess-sovelluskehityksen JavaDoc –dokumentaatio**

<http://www.hut.fi/~tvenho/thesis/javadoc/>